*energies*

MDPI

*Review*

# Towards a Software-Defined Industrial IoT-Edge Network for Next-Generation Offshore Wind Farms: State of the Art, Resilience, and Self-X Network and Service Management

**Agrippina Mwangi [1], Rishikesh Sahay [2], Elena Fumagalli [1], Mikkel Gryning [3] and Madeleine Gibescu [1,*]**

[1] Copernicus Institute of Sustainable Development, Utrecht University, 3584 CB Utrecht, The Netherlands; a.w.mwangi@uu.nl (A.M.) ; e.m.fumagalli@uu.nl (E.F.)
[2] College of Engineering, Technology and Management (ETM), Oregon Institute of Technology, Klamath Falls, OR 97601, USA
[3] Ørsted Wind Power, Skærbæk, 7000 Fredericia, Denmark
* Correspondence: m.gibescu@uu.nl

**Abstract:** Offshore wind farms are growing in complexity and size, expanding deeper into maritime environments to capture stronger and steadier wind energy. Like other domains in the energy sector, the wind energy domain is continuing to digitalize its systems by embracing Industry 4.0 technologies such as the Industrial Internet of Things (IIoT), virtualization, and edge computing to monitor and manage its critical infrastructure remotely. Adopting these technologies creates dynamic, scalable, and cost-effective data-acquisition systems. At the heart of these data-acquisition systems is a communication network that facilitates data transfer between communicating nodes. Given the challenges of configuring, managing, and troubleshooting large-scale communication networks, this review paper explores the adoption of the state-of-the-art software-defined networking (SDN) and network function virtualization (NFV) technologies in the design of next-generation offshore wind farm IIoT–Edge communication networks. While SDN and NFV technologies present a promising solution to address the challenges of these large-scale communication networks, this paper discusses the SDN/NFV-related performance, security, reliability, and scalability concerns, highlighting current mitigation strategies. Building on these mitigation strategies, the concept of resilience (that is, the ability to recover from component failures, attacks, and service interruptions) is given special attention. The paper highlights the self-X (self-configuring, self-healing, and self-optimizing) approaches that build resilience in the software-defined IIoT–Edge communication network architectures. These resilience approaches enable the network to autonomously adjust its configuration, self-repair during stochastic failures, and optimize performance in response to changing conditions. The paper concludes that resilient software-defined IIoT–Edge communication networks will play a big role in guaranteeing seamless next-generation offshore wind farm operations by facilitating critical, latency-sensitive data transfers.

**Keywords:** IIoT-Edge; software-defined networking; network function virtualization; ETSI AI-driven autonomous networks; IEC61850; IEC61400-25; publish/subscribe; resilience; offshore wind farms



**Citation:** Mwangi, A.; Sahay, R.; Fumagalli, E.; Gryning, M.; Gibescu, M. Towards a Software-Defined Industrial IoT-Edge Network for Next-Generation Offshore Wind Farms: State of the Art, Resilience, and Self-X Network and Service Management. *Energies* **2024**, *17*, 2897. https://doi.org/10.3390/en17122897

Academic Editor: Davide Astolfi

Received: 29 April 2024
Revised: 26 May 2024
Accepted: 5 June 2024
Published: 13 June 2024



## 1. Introduction

As coastal nations move to maritime environments to harness stronger and more consistent wind energy, there arises a need for improved remote monitoring, operation, and maintenance. The intricacies of sea environments such as unpredictable weather patterns and the long distance from shore make manual monitoring and maintenance a logistical and economically draining challenge [1]. The offshore wind farm operations and maintenance (O&M) personnel use data-acquisition systems to gather real-time data on critical infrastructure, promptly identifying and resolving issues like equipment malfunctions

or suboptimal performance [2,3]. These data-acquisition systems help minimize prolonged wind farm downtime and maintain peak efficiency in wind farms [4].

Embracing the digital era, the energy sector is undergoing a significant transformation that will see next-generation offshore wind farms leverage the power of technologies from the fourth industrial revolution (Industry 4.0) such as Industrial Internet of Things (IIoT), edge computing, and virtualization to improve their data-acquisition systems. IIoT devices installed on wind turbines and offshore digital substations monitor critical infrastructure in real time. These sensors record wind speed, turbine rotation, temperature, vibration, and electrical output. The data are then transmitted to control and protection systems, enabling automatic adjustments such as altering turbine blade pitch for optimal wind capture or shutting down turbines during severe weather conditions such as storms [5,6].

These large-scale, complex offshore wind farms generate big data from IIoT devices, often managed by cloud services in an IoT/Cloud-based model, as described in [7]. The traditional IoT/Cloud model encounters challenges such as high latency, substantial data transfer, storage subscription expenses, intricate scaling processes, bandwidth limitations, and restricted connectivity [8]. To address this, edge computing is integrated into the design of offshore wind farm data-acquisition systems, bringing storage and computing resources nearer to the assets to improve efficiency [9,10]. Furthermore, software-based protection and control systems are set to replace conventional hardware to centralize management and enable remote monitoring, streamlining fault response, and maintaining wind farm stability [11]. These virtual solutions allow for swift adaptations, upgrades, and new features with minimal physical intervention, cutting down on wind farm downtime and costs [12].

A high-performance, secure, reliable, and scalable communication network is needed to facilitate critical, latency-sensitive data transfer between communicating nodes in the data-acquisition system [13,14]. This communication network is implemented between the wind turbines and an offshore platform (which houses the digital substation). High performance ensures swift data transfer with minimal latency, crucial for real-time wind farm monitoring [15]. Security is upheld through robust encryption and vigilant network monitoring, protecting against unauthorized access and cyber threats. Reliability is achieved with failover systems and durable infrastructure, enabling consistent operation under adverse conditions [16]. Scalability allows for future expansion, accommodating more turbines or increased data loads without compromising network integrity. Together, these attributes ensure that the network can effectively support critical operations of the offshore wind farm.

Over the years, communication networks have significantly improved, adopting new network and service management strategies. Figure 1 shows the evolution of network automation for network and service management from 1960, the launch year of the first network, to 2050. Conventional communication networks have a distributed architecture where each device is managed separately in a concept known as *"siloed networking"*. However, configuring, managing, and troubleshooting have become increasingly challenging as communication networks expand. To address these challenges, researchers continue to explore the concept of *software-defined networking*, where the control plane of the network is decoupled from the data plane [17,18]. This transforms conventional communication networks into programmable, software-defined, and centrally managed architectures, improving network management and agility, streamlining the deployment of new services, and reducing reliance on vendors [19,20]. These technologies enable more efficient data traffic routing, improved system scalability, and greater flexibility in responding to dynamic network demands, all while potentially lowering operational costs and simplifying maintenance procedures [21].

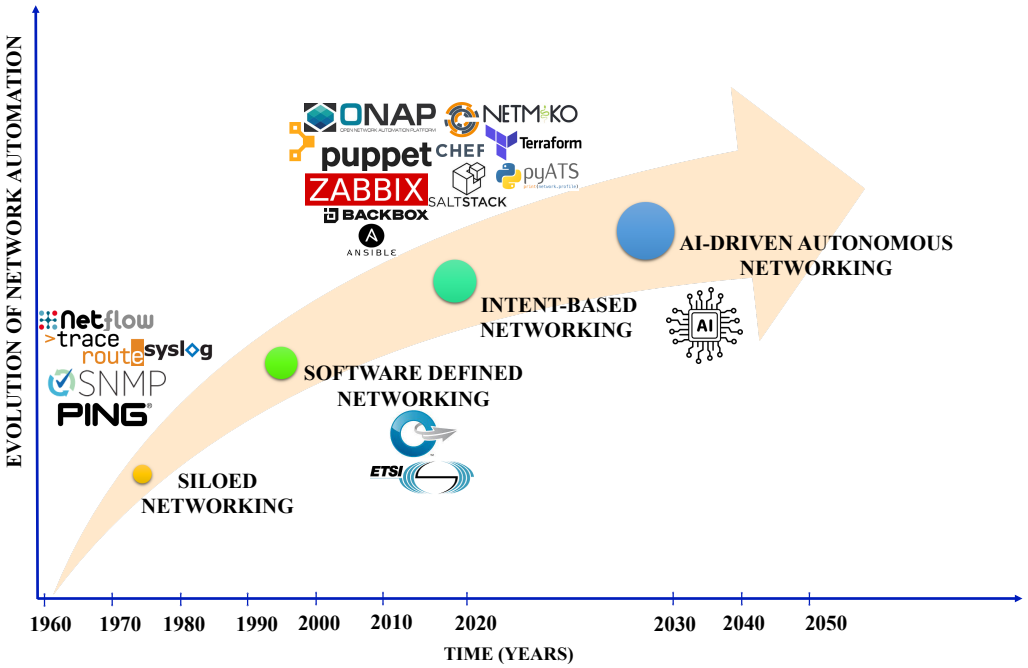


**Figure 1.** The evolution of network automation for network and service management (1960 to 2050 and beyond).

While these software-defined, programmable networks present myriad benefits, they face a new set of performance, security, reliability, and scalability challenges. To tackle these challenges autonomously while ensuring that the industrial service level agreements are met, organizations like the European Telecommunication Standards Institute (ETSI) and the Open Network Foundation (ONF) have introduced the concept of *intent-based networking (IBN)*. IBN aligns network operations to meet the specific business goals or *"intents"* of the application (in this case wind farm operation), both initially and for future changes [22]. Intents are declarations of desired outcomes or operational states. These intents are translated into network configurations and policies using an abstraction module. IBNs continuously monitor the current state to ensure it aligns with the defined intents [23]. Future networks will use *"AI-driven autonomous networking"* concepts where an AI/ML approach is deployed in the network to predict its behavior and suggest optimization techniques to self-configure, self-heal, and self-optimize these networks. When performance anomalies or security breaches occur, these AI-driven autonomous networks trigger self-healing modules to correct faults without human intervention.

### *1.1. Significance and Contributions*

The significance of this study lies in its potential to change the design and operation of large-scale communication networks to the offshore wind farm application scenario. This review addresses a critical gap in the literature concerning the adoption of SDN/NFV-based architectures in the design of offshore wind farm Operational Technology (OT) communication networks with stringent Quality of Service (QoS) requirements. The paper serves as a guiding beacon for practitioners seeking to understand the latest advancements in this communication network domain. Also, it provides researchers with valuable insights to identify and pursue promising research streams. Subsequently, this review paper:

- Conducts a comprehensive review of how Industry 4.0, IIoT, Edge computing, and virtualization technologies will be integrated into next-generation offshore wind farm data-acquisition systems design,

- Examines the performance, security, reliability, and scalability challenges of implementing software-defined networking and network function virtualization in the design of IIoT–Edge networks,
- Discusses approaches to mitigate the highlighted challenges and build resilience in the next-generation offshore wind farm's software-defined IIoT–Edge networks.

### *1.2. Organization of the Paper*

The rest of the review paper is organized as follows: Section 2 explores the integration of Industry 4.0 IIoT and Edge computing in the design of data-acquisition systems for next-generation offshore wind farms. Further, Section 3 discusses the adoption of SDN and NFV in the design of the communication network that facilitates critical, latency-sensitive data transfer in the IIoT–Edge data-acquisition system. The performance, security, reliability, and scalability concerns of SDN/NFV-based communication networks are addressed in Section 4. Potential mitigation strategies and self-X network and service management approaches to build resilience in this network are discussed in Section 5. The limitations of the study are highlighted in Sections 6 and 7 concludes the paper.

## 2. Leveraging Industry 4.0 IIoT and Edge Computing in Next-Generation Offshore Wind Farms Data-Acquisition System Design

### *2.1. Overview*

IIoT devices facilitate proactive monitoring, condition-based maintenance, automation, alerts, scheduling, forecasting, and predictive analysis, which help the O&M team in decision-making [5,24]. Surveys by Mustafa et al. [25] and Wisser et al. [26] show that of the total levelized cost of energy (LCOE) in a wind farm, 49% is attributed to the wind turbine energy generation, 29% to O&M functions, 16% to auxiliary systems, and 6% to regular financial costs. To reduce the 29% LCOE attributed to O&M functions, wind farm operators are gradually adopting IEC 61400-25 [27] industry-grade Internet of Things (IIoT) devices alongside other customized IIoT devices to monitor their critical infrastructure [28,29]. In turn, this significantly reduces the high O&M costs.

Some common commercial wind turbine monitoring platforms are Windmill Manager (WebNMS), which supports condition-based wind turbine performance monitoring and maintenance; Cloud-based IoT Solution (Qburst), which supports wind turbine parameter monitoring, alerts, and calendar-based maintenance management; Digital WindFarm (General Electric), which supports maintenance strategies, reliability, and availability assessment; EnOS™ Wind (Envision), which supports remote regional supervision and maintenance; and MindSphere (Siemens AG innovation), which connects IIoT devices to a cloud platform, harnessing big data from offshore wind farm data-acquisition systems and enhancing operations through the creation of digital twins [30].

### *2.2. Industry 4.0 IIoT, Cloud, and Edge Computing for Next-Generation Offshore Wind Farms*

IIoT devices, mounted on wind turbines within the wind farm, communicate with each other to exchange environmental data and coordinate actions in the wind farm to prevent damage or accidents. The interaction between wind turbines and remote control centers is crucial for efficient operation and management. Key activities in this interaction are (i) the transmission and reception of alarms, (ii) the initiation of service requests, (iii) the adjustment of control parameters, and (iv) the collection of turbine status and performance data [31]. The implementation of these IIoT devices for monitoring and maintenance significantly enhances these processes and leads to several benefits for wind farm operators, including the prevention of extended downtime, the reduction in operation and maintenance (O&M) costs through the early identification of faults, and the enhancement of real-time decision-making capabilities [32]. These improvements not only provide direct financial benefits but also contribute to indirect economic advantages. To quantify these benefits, Zhou et al. [33] developed a maintenance cost model that showed

up to 75% reduction in annual maintenance costs and downtime ratio in wind power equipment's IIoT-enabled offshore wind farms.

An IIoT system comprises IIoT devices, a communication network, and cloud storage servers (with advanced computation capabilities). Figure 2 illustrates the seven-layer IoT World Forum Reference Model adapted from [34], which illustrates all the layers of an IIoT system. The reference model layers are grouped into the Data-in-Motion (DIM) module, which comprises Layer 1 to Layer 3, and the Data-at-Rest (DAR) module, which comprises Layer 5 to Layer 7. For an offshore wind farm application scenario, the DIM module comprises real-time, operational technology (OT), and event-based functions of the offshore wind farm. Here, the functions and flow of data are based on the events of the application domain. The DAR module consists of non-real-time business applications implemented as software. Further, the module's end users or applications run queries on the data to capture only data relevant to their particular interests (predictive analytics, forecasting, trends, and visualization).

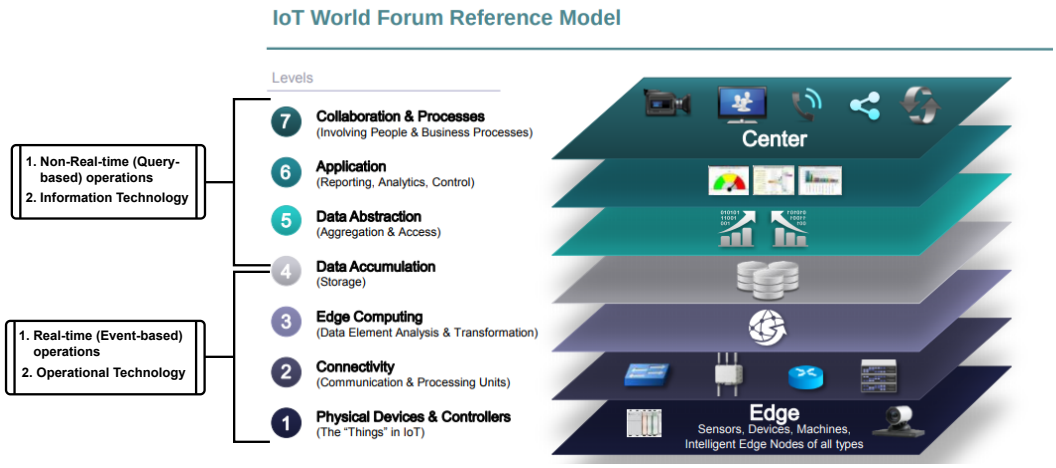


**Figure 2.** The seven-layer Internet of Things World Forum reference model, adapted from [34] and classified into Data-in-Motion (Layer 1 to Layer 3) and Data-at-rest (Layer 5 to Layer 7).

The IoT World Forum reference model layers are described as two modules DIM and DAR:

**Data-in-Motion (DIM) Module:** The Data-in-Motion module comprises layer 1 to layer 3. Layer 1 (Physical Devices and Controllers): Vibration, environmental, and operational IIoT devices are mounted to monitor critical components in an offshore wind farm [35]. Considering their application environment, these IIoT devices are ruggedized to withstand harsh environmental conditions such as extreme temperatures, humidity, vibrations, and exposure to dust and chemicals. They are periodically recalibrated to provide precise and reliable data consistently over time. Layer 2 (Connectivity): The IIoT devices send the digitized measurements through communication and processing units to the edge computing platform [36]. Several wired and wireless technologies are deployed in this layer to facilitate latency-sensitive data transfer within short and long ranges. The communication technologies adopted at this layer are either wired (such as fiber optic cable, ethernet copper cable, and coaxial cable) or wireless (such as NB-IoT, Wi-Fi, LoRA, LTE, WiMax, Bluetooth, SigFox, NFC) solutions [37]. Layer 3 (Edge Computing): The measurements are received at the edge computing layer, which provides advanced computing and memory resources similar to those offered by a cloud but closer to the physical devices and controllers in layer 1 [38].

**Layer 4 (Data Accumulation):** The layer stores and manages the big data obtained from the physical devices and controllers in Layer 1. The offshore wind farm's data-acquisition system's attributes are variety, which *"refers to the different types of structured and unstructured data such as text, sensor data, audio, video, and graph from heterogeneous data sources"* [39]; velocity, which *"refers to how quickly the data can be analyzed to make decisions or*

*to obtain useful analytic information in real-time"* [40]; veracity, which refers to *"the reliability and insightfulness of data"* [41], and volume, which *"refers to the large amount of data that grows continuously as more data is obtained from the IIoT devices mounted on the offshore wind farm's critical infrastructure"* [40].

**Data-at-Rest (DAR) Module:** The data-at-rest (DAR) module comprises Layer 5 to Layer 7. Layer 5 (Data Abstraction): This layer aggregates the data and simplifies how the underlying data are represented, abstracting the end users from unnecessary background details of the heterogeneous data using object-oriented programming and other database management tools [40,41]. Layer 6 (Application): This layer comprises software applications built on the microservices architecture that performs monitoring, visualization, predictive analytics, forecasting, trends, and control logic using application programming interfaces to interact with the data stored in Layer 7 [39]. Layer 7 (Collaboration and Processes): This is the business layer, where the business objectives, key performance indicators (KPIs), and other requirements are defined [34].

Contemporary IIoT systems use the IoT–Cloud model, where IIoT devices collect data from the critical infrastructure under observation and send them through a wide area network (or the Internet) to the cloud instances [42]. Then, using query-based approaches, the data stored in the cloud storage are accessed by the O&M team for visualization, forecasting, pattern recognition, and predictive analytics. Regrettably, this IoT–Cloud model faces several technical, operational, and regulation challenges [43]. The continued proliferation of these IIoT devices creates a huge data influx, testing the resilience of the communication infrastructure by increasing the bandwidth requirements [44]. As such, the IoT–cloud model experiences a significant response time lag between the IIoT devices sending data to the cloud instance, the cloud processing the data, and the response being sent to the wind farm. This response time lag is further aggravated when the data are sent over the Internet, which is a best-effort network [8,45]. Further, adding more devices increases the task load at the central cloud infrastructure, causing scalability concerns [46]. Additionally, many cloud servers are located in different regions and adhere to diverse data protection and privacy regulations [47]. As such, the model faces data sovereignty and compliance issues.

Conversely, the IIoT–Edge model enables processing data near the IIoT devices, thereby reducing latency, conserving bandwidth, bolstering reliability, and significantly strengthening security. Cao et al. [2] conducted an experimental simulation to demonstrate how leveraging edge computing technology in wind farm data-acquisition systems design eases acquisition and makes its management more efficient and accurate. Further, Zhang et al. [9] performed structural health monitoring on offshore wind turbine towers by measuring the acceleration response, incorporating an edge-computing framework in its data-acquisition system. Xu et al. [10] designed a multi-sensor edge computing architecture to identify incipient short-circuits in wind turbine electrical generators. The results from these empirical studies quantitatively demonstrate how processing data close to the wind farm improves offshore wind farm operations. By enabling real-time data processing and immediate response capabilities, edge computing enhances operational efficiency, improves safety, and reduces maintenance costs [48]. These benefits are crucial for the sustainable operation of wind farms and underscore the growing importance of integrating advanced computational technologies in renewable energy systems.

Designing scalable IIoT–Edge data-acquisition systems using the publish/subscribe model allows a form of asynchronous messaging where the publishers or data producers are decoupled from the subscribers or data consumers [49]. In this model, publishers send messages to a message broker, which stores messages in topics [50], and the subscribers then subscribe to these topics. Depending on the requirements of the wind farm operator, it is likely that they may add or remove publishers or subscribers from the IIoT–Edge data-acquisition system illustrated in Figure 3. This model excels in scalability for wind farms' data-acquisition systems, as it allows the number of publishers to be changed

without impacting the subscribers, due to its non-reliance on a 1:1 publisher-to-subscriber mapping [46].

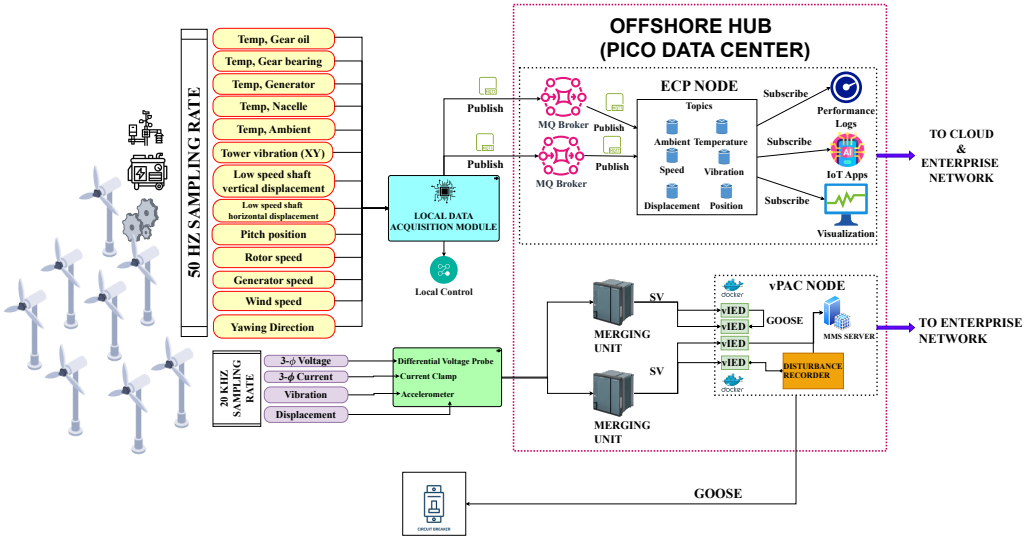


**Figure 3.** Architecture of a next-generation offshore wind farm data-acquisition system leveraging Industry 4.0 IIoT, Edge, and virtualization technologies using a publish/subscribe model.

There are key IoT protocols that support the publish/subscribe model for machine-to-machine communication in industrial IoT environments. Elhadi et al. [51] quantify the adoption of these IoT protocols at the IoT-cloud platform level with MQTT at 31%, AMQP at 19%, COAP at 13%, and OPC-UA at 6%. The choice of IoT protocol emerges as an important challenge for application developers who must choose one of the many lightweight communication protocols for a resource-constrained environment: ISO 19464 Advanced Message Queueing Protocol (AMQP), also known as "Internet protocol for business messaging" is an open standard lightweight message-based protocol that assists cross-platform application communication. It supports the publish/subscribe model alongside the request/respond model and the store-and-forward model [52,53]. The AMQP supports asynchronous communication by connecting its brokers and applications in a star design. Further, it integrates quality of service (QoS) features, requiring that all applications know the broker's URL. Unfortunately, AMQP requires more bandwidth resources than its counterparts and may not be suitable in resource-constrained environments.

IBM's OASIS standard Message Queueing Telemetry Transport (MQTT) is designed with a small code footprint with minimal bandwidth requirements, making it suitable for deployment in low-power, resource-constrained machine-to-machine communication scenarios [54,55]. Unlike AMQP, MQTT considers multiple QoS levels (such as QoS level 1: *"fire-and-forget"*; QoS level 2: *"at least once"*; and QoS level 3: *"exactly once"*) [53]. Further, it is quite scalable and finds applications in scenarios with several small, constrained devices with more reliability requirements than speed [56]. Common examples of MQTT in offshore wind include the use of the MQTT Sparkplug in optimizing SCADA implementations.

Internet Engineering Task Force (IETF) RFC-7252 Constrained Application Protocol (CoAP) is a specialized User Datagram Protocol (UDP) web transfer protocol for use with constrained nodes and constrained networks in industrial IoT environments such as advanced metering in digital substations in the smart energy domain and building automation [7,51,57]. Some common commercial testbeds running CoAP are InterDigital oneMPOWER, ARM mbed, and thethings.io.

IEC 62541 Open Platform Communication Unified Architecture (OPC-UA) is a platform-independent, broker-less, service-oriented architecture that supports pub/sub communication modes for time-sensitive networking and UDP multicast application scenarios. In implementation, the publish and the subscribe servers are successfully decoupled to realize point-to-multipoint transmission [58].

Figure 3 illustrates a data-acquisition system designed in the IIoT–Edge computing paradigm. In the figure, IIoT sensors are mounted to measure physical quantities on critical infrastructure. These IIoT devices collect highly granular data at different sampling rates. The data collected from the wind turbines is sent via sub-sea ethernet-based fiber optical cables to the offshore hub (set to house a pico data center (PDC)). A cluster of x86 servers, with diverse computing and memory capacities, are deployed within the PDC. Using the server cluster applications, these computing and memory resources are clustered in a unified server resource pool [32,59] and allocated to the Edge Computing Platform (ECP) node, virtual Protection, Automation, and Control (vPAC) node, and other IT/OT systems needed to facilitate wind farm operations.

Table 1 denotes the data transfer performance requirements and communication modes for different offshore wind farm services. The communication direction is based on the IT/OT architecture of the next-generation offshore wind farm data-acquisition system illustrated in Figure 3. These offshore wind farm services are further described in the following two data transfer scenarios:

**IIoT-to-ECP node scenario:** The IEC61400-25 wind turbine and metmast sensors collect highly granular analog measurements at a 50Hz sampling rate and send them to a local data acquisition module [2] within the wind turbine, as illustrated in Figure 3. This local data acquisition module pre-processes these data and facilitates actuation through the local control ensemble in the wind turbine. Further, it sends the sensor data through short-range ethernet-based fiber optic patch cords to the nacelle switch [4]. The nacelle switch communicates with the tower switch, sending the data as ethernet frames to the PDC switch network through the multiplexed subsea fiber optic cable. The PDC switch network then relays the data to the ECP node's MQTT broker [60]. The sensor data received at the MQTT broker are stored in different topics. In this design, it is inferred that some topics contain sub-topics. For instance, the "*/wind-turbine/temperature*" topic contains the subtopics "*/wind-turbine/temperature/gear-oil/*", "*/wind-turbine/temperature/nacelle/*", "*/wind-turbine/temperature/ambient/*", and others, structured hierarchically for efficient data categorization and retrieval. In the ECP node, IoT apps subscribe to the data in the topics using the Apache Kafka message streaming tool to manage high subscription rates. Additionally, the cloud instance applications and the O&M team (enterprise network) subscribe to these data through the wide area network.

**IIoT-to-vPAC node scenario:** Current and voltage sensors are mounted on critical electrical components in the wind turbine to collect analog current and voltage measurements at a 20 kHz sampling rate [12]. As illustrated in Figure 3, these current and voltage sensors are connected to differential voltage probes and current clamps. The differential voltage probe and the current clamp send the analog current and voltage measurements to merging units (MU) using short-range ethernet-based fiber optic cables. The MU digitizes these analog current and voltage measurements into IEC 61850-9-2 Sampled Values (SVs) — according to the SV protocol, as described at https://www.typhoon-hil.com/documentation/typhoon-hil-software-manual/References/iec_61850_sampled_values_protocol.html (accessed on 4 June 2024) — and then publishes them on the process bus network. At the vPAC node, virtual intelligent electronic device (vIED) docker-based containers, designed using relay technology, are configured and deployed in the docker engine. The Kalkitech's virtual protection relay (VPR) reference system affirms that using vIEDs in a dockerized environment meets and exceeds the vIED to vIED performance and time requirements [11]. Additionally, the environment facilitates redundancy, resiliency through container migration, and ease of deployment [61]. These vIEDs subscribe to the SVs and process the data. The vIEDs send the processed SVs to the disturbance recorder (a containerized or VM instance) to determine the power quality by detecting transients or short-duration voltage variations in the electrical network under observation. Additionally, these vIEDs exchange IEC 61850 Generic Object-Oriented Substation Events (GOOSE) messages — according to the GOOSE protocol, as described at https://www.typhoon-hil.com/documentation/typhoon-hil-software-manual/References/iec_61850_goose_protocol.html (accessed on 4 June 2024) — with each other through a virtual switch and with external process-level

equipment such as circuit breakers through the process bus network [62]. Lastly, these vIEDs communicate with other station-level equipment by sending IEC 61850 Manufacturing Message Specification (MMS) signals to the historian database in the enterprise network or other energy management systems in the cloud control center.

**Table 1.** Offshore wind farm data communication parameters between the wind turbine generators (WTG), the offshore hub-mounted pico datacenter components, and the Internet [20].

| Service | Communication Direction | Priority | Data Rate | Latency | Reliability | Packet Loss Rate |
|---|---|---|---|---|---|---|
| Protection traffic | WTG → vPAC | 1 | 76,816 bytes/s | 4 ms | 99.999% | $<10^{-9}$ |
| Analogue measurements | WTG → vPAC/ECP | 2 | 225,544 bytes/s | 16 ms | 99.999% | $<10^{-6}$ |
| Status information | WTG → ECP | 2 | 58 bytes/s | 16 ms | 99.999% | $<10^{-6}$ |
| Reporting and logging | WTG → ECP | 3 | 15 KB every 10 min | 1 s | 99.999% | $<10^{-6}$ |
| Video surveillance | WTG → ECP | 4 | 250 kb/s–1.5 Mb/s | 1 s | 99% | No specific requirement |
| Control traffic | vPAC → WTG | 1 | 20 kbs/per turbine | 16 ms | 99.999% | $<10^{-9}$ |
| Data polling | ECP/vPAC → WTG | 2 | 2 KB every second | 16 ms | 99.999% | $<10^{-6}$ |
| Internet connection | Internet → WTG/ECP/vPAC | 3 | 1 GB every two months | 60 min | 99% | No specific requirement |

## 3. Software-Defined Networking (SDN) and Network Function Virtualization (NFV) for Next-Generation Offshore Wind Farms

Central to the next-generation offshore wind farm's IIoT–Edge data-acquisition system (shown in Figure 3) is a communication network that leverages SDN and NFV technologies to facilitate critical, latency-sensitive data transfer [63,64]. There is a need to model robust communication frameworks to guarantee seamless offshore wind farm operations [65].

Conventionally, wind farm developers source their networking equipment from one or more vendors to support various network functions [66]. These communication networks, referred to as traditional networks in this paper, run distributed architectures to support industrial OT network operations. While these traditional networks have dominated the current designs of large-scale communication networks in offshore wind farms and other industrial OT networks, they face several technical and logistical challenges. These traditional network architectures incur huge cost implications and are quite complex to manage when more devices, users, and network traffic volumes increase [67]. Further, configuring and managing these networks involves individual-component manual configurations of individual devices, increasing the risk of configuration errors as the network enlarges [68]. These networks face vendor lock-in for cases where the wind farm developer relies heavily on proprietary hardware and software from a single vendor. Additionally, this vendor lock-in limits innovation such that network administrators cannot customize the network functionalities to suit their business intents outside the provisions of the vendor [69,70]. Given the siloed network management approach, there is limited global network visibility and control; hence, it is challenging to troubleshoot issues or optimize network resources [71]. Most importantly, these networks have a longer convergence time, referring to the time the network takes to recover and resume normal operations after a fault such as a link or a device failure [67]. Table 2 shows a feature-based comparison between the traditional communication networks and the software-defined networks. Figure 4 illustrates the distributed architecture of the traditional networks against the centralized architecture of the software-defined networks.

**Table 2.** A feature-based comparison between the traditional communication networks and the software-defined networks.

| Features | Traditional Networks | Software-Defined Networks |
|---|---|---|
| Architectural Design | Distributed design with control plane and data plane coupled in a single device | Centralized design with decoupled control plane and data plane |
| Programmability | Non-programmable;<br>Difficult to replace existing program as per use | Programmable;<br>Easy to update existing program per use |
| Configuring and Managing | Supports static and manual configuration<br>Difficult to troubleshoot and report in a distributed control design | Supports reactive/proactive automated configuration<br>Easy to troubleshoot and report in a centrally controlled design |
| Cost implications | High CAPEX and OPEX | High CAPEX and low OPEX |

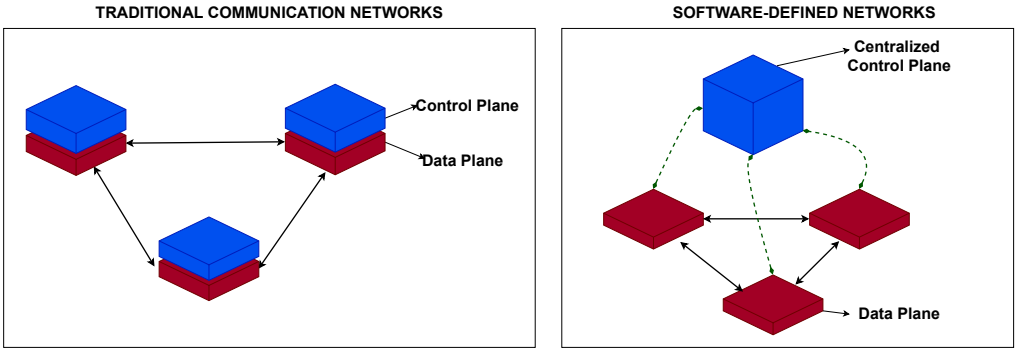


**Figure 4.** Comparing the traditional networks with the software-defined networks applicable to both switch and router networks.

### *3.1. Leveraging Software-Defined Networking (SDNs) in the Design of IIoT–Edge Networks*

The term "*Software-defined networking*" was coined by Kate Green in *MIT Technology Review* in 2009 when describing the newly created OpenFlow specifications [72] as "*a networking approach that utilizes software-based controllers or application programming interfaces (APIs) to manage hardware infrastructure and direct network traffic*". Other researchers have adopted the term to represent their ideas and work around OpenFlow [73–75]. Fundamentally, a software-defined network (SDN) decouples the control plane from the data plane, making the network devices (in the data plane) programmable through standardized application interfaces. It differs from traditional networks where the control and data planes are bundled in a single device.

The control plane makes the forwarding decisions, packages them as flow instructions, and sends them asynchronously to the data plane that comprises ethernet switches known as forwarding devices (FDs) [75,76]. These FDs forward the packet from source to destination. Figure 5 illustrates the SDN architecture, comprising four planes: (i) data plane, (ii) control plane, (iii) application plane, and (iv) management plane [18,77]:

**Data Plane:** The data plane, at the bottom of the architecture, comprises the FDs that forward ethernet packets or frames from source to destination. These FDs in the data plane rely on the controllers in the control plane to determine their forwarding behavior. As such, the data plane FDs interact with the control plane's controllers via the Southbound Interface (SBI) using the OpenFlow protocol [73]. At the SBI, the other protocols used besides OpenFlow protocol are Open vSwitch Database (OVSB) [78], Protocol-Oblivious Forwarding (POF) [79], CISCO OpFLEX [80], OpenState [81], Revised Open-Flow Library (ROFL) [80], Hardware Abstraction Layer (HAL) [73], programmable abstraction of data path (PAD) [82], and Forwarding and Control Element Separation (ForCES) [83].

**Control Plane:** To ensure that the control plane is not a single point of failure in the architecture, $(n+1)$ controllers are deployed as either a flat distributed peer-to-peer or a hierarchical-based cluster. They communicate with each other, sharing the network state of the FDs in the data plane under their administrative domain through an east/westbound interface [80,84]. The controller's core functions include, but are not limited to, manag-

ing the network topology, link discovery, collecting network-based statistics, managing queues, and traffic flow [85]. Currently, there exists a wide array of SDN controllers such as OpenDayLight (ODL) [86], Open Network Operating System (ONOS) [87], Python-based SDN Openflow eXperimentation platform (POX) [88], Floodlight [89], Network Operating System (NOX) [90], Ryu [91], Beacon [92], Maestro [92], Iris [93], MUL [84], Runox [84], SEL-5056 SDN flow controller [94], and Lib-Fluid [84]. These SDN controllers are designed using different programming languages (JAVA, C++, Python), are compatible with different SBI and NBI application programming interfaces (APIs), are developed by different industry partners, and can be used in different application scenarios (see Table 3). Isong et al. [95] categorize common SDN controllers (both open-source and proprietary) as (i) physically centralized (NOX, NOX-MT, POX, Maestro, Ryu, Floodlight, and Beacon), (ii) physically flat-distributed (ONIX, HyperFlow, OpenDayLight, ONOS, DISCO), (iii) physically hierarchically distributed (Kandoo, Orion, D-SDN, B4, Espresso), (iv) logically centralized (POX, Ryu, ONIX, SMaRtLight, HyperFlow, Ravana, OpenDayLight, ONOS, B4, SWAN, Espresso), and logically distributed (DISCO, D-SDN, Cardigan, SDX, ISDX, ToulX, AtlanticWave).

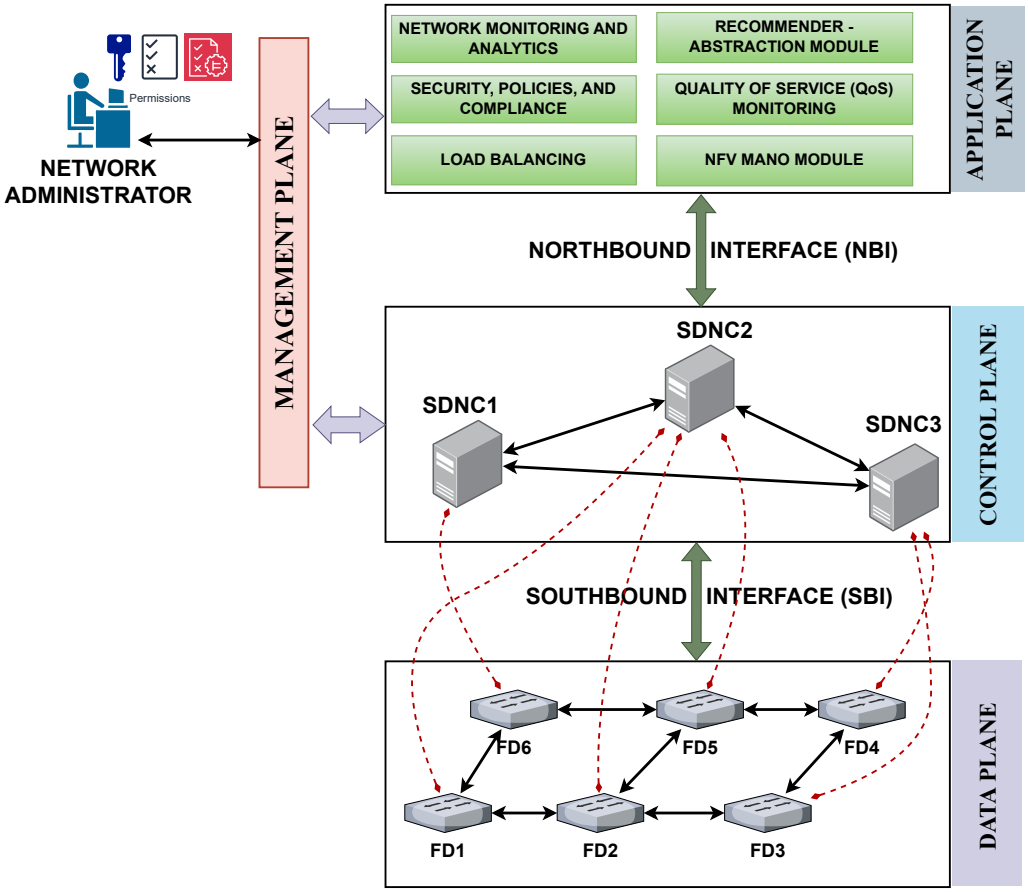


**Figure 5.** Software-defined networking (SDN) architecture, where SDNC is the SDN controller at the control plane and FD is the forwarding device at the data plane.

**Application Plane:** The application plane hosts Representational State Transfer (REST)-based network applications and services such as network monitoring and analytics, security and compliance, load balancing, and traffic engineering that monitor the network's quality of service metrics [96,97]. The control plane's controllers interact with these REST-based applications via the Northbound Interface (NBI) using the RESTCONF protocol [98,99]. Other techniques used at the NBI are RESTful APIs, OpenFlow v1.3-1.5, NETCONF, Google Remote Procedure Call (gRPC), and custom APIs and protocols.

**Management Plane:** The management plane allows the network administrators to securely (via Secure Shell (SSH) or Telnet) access the application plane or control plane directly to make policy and flow modifications as and when necessary.

Unlike traditional networks, vendor-agnostic SDN technology is increasingly being adopted in the design of industrial networks with several benefits. Decoupling the control plane from the data plane provides dynamic centralized control as the controllers deployed in the control plane have a global view of the network [73]. Further, with a global view, the controller can quickly detect faults in links and devices in the data plane and send alerts to the network administrator or autonomous applications running in the application plane [99]. Configuration and traffic engineering tasks can be automated, facilitating a simpler configuration and management approach that can be rolled back [80]. The network is programmable, giving the network administrators leverage to modify their networks to meet their customized intents and achieve the industrial service level agreements (SLA) or key performance indicators of the network [18]. There is reduced vendor lock-in as SDN can effortlessly manage a heterogeneous data plane comprising devices from different vendors [77]. Lastly, and most importantly, SDN achieves faster convergence times (of up to 100 µs) than traditional networks (10–30 ms) because it can rapidly reconfigure network paths and forwarding behavior in response to faults or changes in network conditions [67].

**Table 3.** Comparison of open-source SDN controllers based on common architectural and operational design features.

| Sources | Controller | Organization | Programming Language | Open Source | Flow/Second | Modularity | Productivity | Consistency | Fault | Architecture |
|---|---|---|---|---|---|---|---|---|---|---|
| [85,87] | ONOS | ON.Lab | Java | Yes | 1M | High | Fair | Weak - Strong | Yes | Flat-distributed |
| [86] | OpenDayLight | Linux Foundation | Java | Yes | 106K | High | Fair | Weak | No | Flat-distributed |
| [89] | Floodlight | Big Switch Network | Java | Yes | - | Fair | Fair | No | No | Centralized Multi-threaded |
| [73] | ONIX | Nicira Networks | C Python | Yes | 2.2M | Fair | Fair | Strong | Yes | Distributed |
| [92] | Beacon | Stanford University | Java | Yes | 12.8M | Fair | Fair | No | No | Centralized Multithreaded |
| [77,100] | Hyperflow | University of Toronto | C++ | | 30K | Fair | Fair | Weak | Yes | Distributed |
| [84,85] | Maestro | Rice University | Java | Yes | 4.8M | Fair | Fair | No | No | Centralized Multithreaded |
| [84] | OpenMUL | KulCloud | C | Yes | | Fair | Fair | No | No | Centralized Multithreaded |
| [91] | RYU | NTT | Python | Yes | | Fair | Fair | No | No | Centralized Multithreaded |
| [88] | POX | Nicira | Python | Yes | 1.8M | Low | Fair | No | No | Centralized |
| [90] | NOX | Nicira | Python | Yes | 1.8M | Low | Fair | No | No | Centralized |

Choosing the best-suited SDN controller (as listed in Table 3) to deploy at the control plane has proven challenging for researchers and network administrators. Several empirical studies used proof-of-concept simulation environments and OpenFlow Protocol Layer frameworks, such as Cbench, to evaluate SDN controllers based on network performance benchmark metrics such as latency, jitter, throughput, and packet loss. To be specific, Cbench is a stress-testing tool from the OpenFlow Protocol Layer for Investigating Performance of SDN (OFLIPS) framework, PktBlaster and OFNet [101]. These frameworks and tools were used to compare the SDN controllers based on the following benchmarks.

**For the latency and response time benchmark:** Experimental studies evaluate the time it takes for each SDN controller to process and respond to the network requests. Salman et al. [84] ranked the Maestro SDN controller as the best performer, having compared it against a wider array of SDN controllers, namely ODL, Floodlight, Beacon, Maestro, IRIS, Ryu, POX, NOX, ONOS, MUL, and Libfluid-based controllers. Khattak et al. [86] observed that ODL, despite experiencing memory leakages, had a shorter response time than Floodlight. Mamushiane et al. [102] observed that Ryu and ODL outperformed the ONOS and Floodlight SDN controllers. Islam et al. [103] compared Ryu, POX, ONOS, and Floodlight in an emulated Mininet–Wifi simulation wireless network. An analysis of the four SDN controllers revealed notable differences in performance. Floodlight exhibited superior performance compared to the others, showcasing the low-

est delay and jitter. Conversely, Ryu displayed the poorest results in jitter tests, while ONOS exhibited the worst delay performance. Moreover, all controllers performed similarly well in throughput tests. Lastly, Zhu et al. [101] qualitatively compared nine SDN controllers, finding that multi-threaded and centralized (Floodlight, OpenMUL, Beacon, Maestro) and distributed (e.g., ODL, ONOS) SDN controllers outperformed the centralized and single-threaded ones (e.g., NOX, POX, Ryu).

**For the throughput benchmark:** Experimental studies have assessed the SDN controller's ability to efficiently handle and manage data traffic. According to Salman et al. [84], MUL and Libfluid-based SDN controllers outperformed ODL, Floodlight, Beacon, Maestro, IRIS, Ryu, POX, NOX, and ONOS. Further, Mamushiane et al. [102] and Singh et al. [104] observed that Ryu and ONOS outperformed ODL and Floodlight.

**For the architectural and network load benchmark:** Controllers have four key architectural features: multicore support, switch partitioning, packet batching, and packet processing. Shah et al. [92] compared four leading open-source SDN controllers (NOX, Beacon, Maestro, and Floodlight), analyzing their architectural designs. They concluded that controllers aiming for high throughput should use static switch partitioning and packet backing. Controllers focused on delay-sensitive applications should employ adaptive packet and task batching to minimize per-packet latencies. Additionally, sending each control message individually can further enhance latency performance. Further, Rowshanrad et al. [97] compared ODL against Floodlight using the Mininet emulator and concluded with a 95% confidence interval that ODL outperforms Floodlight in low-loaded networks and tree topologies in mid-loaded networks in terms of latency. However, they also concluded that Floodlight can outperform ODL in heavily loaded networks for tree topologies in terms of packet loss and linear topologies in terms of latency.

**For the scalability benchmark:** Mamushiane et al. [102] observed that ONOS outperformed ODL, Ryu, and Floodlight and were affected by increased workloads. They also noted that SDN controller placement affects performance across various network topologies. Currently, researchers are experimenting with customized SDN controllers specific to certain applications. Zhu et al. [85] propose deploying customized SDN controllers that can accommodate protocol interpreters for IoT application scenarios.

Drawing upon the findings of the experimental studies discussed, the choice of an SDN controller should be contingent upon specific application requirements, given the diverse advantages and disadvantages observed among the SDN controllers evaluated. This necessitates the careful consideration of trade-offs to align with the intended application environment's operational demands and performance expectations.

### *3.2. Leveraging Network Functions Virtualization (NFV) in the Design of IIoT–Edge Networks*

According to the ETSI, Network Function Virtualization (NFV) transforms network services that are traditionally implemented as hardware-based appliances [105] into Virtual Network Functions (VNF) deployed as Virtual Machines (VMs) or containers. NFV tackles the challenge of hardware-based appliances nearing their end of life due to accelerated technology and innovation [105,106]. Hardware-based appliances such as firewalls, load balancers, Virtual Private Network (VPN) gateways, Deep Packet Inspection (DPI), Session Border Controllers (SBC), carrier-grade Network Address Translator (NAT), Quality of Experience (QoE) monitor, and Wide Area Network (WAN) acceleration are hosted on standard industrial server hardware [105,107]. There are several benefits of adopting ETSI NFV in the IIoT–Edge network:

- **Cost Reduction:** Capital and operational expenses are reduced with reduced hardware-based appliance deployment and reduced power consumption.
- **Scalability and Flexibility:** It facilitates dynamic scaling of VNF VMs or container instances to meet the demand. Further, these VMs and containers can be instantiated, decommissioned, or migrated providing flexibility to accommodate changes in network traffic volumes and patterns.

- **Efficient Resource Utilization:** NFV VMs and containers optimize resource utilization by consolidating multiple functions onto shared industrial server hardware [108]. Further, these VM and container instances' resource utilization is managed by Kubernetes or other proprietary or open-source orchestrators.
- **Service Innovation:** NFV enables the rapid introduction and deployment of network services. The VNFs can be developed, tested, and deployed in the Continuous Integration/Continuous Delivery (CI/CD) approach, reducing the time-to-market of new services and features.
- **Resilience and Redundancy:** Several VNF instances are deployed on industry-grade servers in multiple locations enabled with automated failover and redundancy mechanisms for load balancing and high availability.
- **Vendor-Agnostic:** Wind farm operators can deploy the VNFs from multiple vendors in their OT networks. This reduces vendor lock-in challenges and fosters competition among the vendors, driving innovation and significantly reducing costs.

### *3.3. SDN/NFV-Based Architectures for Industrial OT Networks*

SDN and ETSI NFV are complementary technologies capable of providing one network solution suited for industrial applications that need constant customization and innovation [20,71,109]. Figure 6 illustrates a software-defined IIoT–Edge network, taking the case of the Anholt offshore wind farm. The Anholt wind farm is a large offshore wind farm with an installed capacity of 400 MW. The offshore wind farm has 111 Siemens SWP 3.6-120 wind turbines with a generation capacity of 3.6 MW each. Orsted Wind Power, Denmark, operates this offshore wind farm. Each wind turbine has an access network with a nacelle switch and a tower switch. This results in $\geq$222 Ethernet switches distributed across the wind farm. These switches handle local data aggregation and transmission. Each tower switch within a wind turbine is connected to the PDC via sub-sea fiber optic cables. This setup ensures robust, high-speed data communication between the distributed wind turbines and the central data-processing hub. Additionally, onsite operations and maintenance (O&M) personnel utilize access points for Internet access, facilitating real-time monitoring and control. The offshore hub-mounted PDC is supported by $n$ additional Ethernet switches, strategically deployed to manage and route the substantial data flow from the wind turbines to the PDC. Considering all components, a minimum of 250 Ethernet switches are involved in forming the IIoT–Edge network for the wind farm. Within the PDC, NFV instances supplement network functions and enhance the security, efficiency, and management of the network.

These SDN/NFV-based architectures are anticipated to be progressively integrated into industrial OT networks due to their multifaceted advantages summarized into four pillars (interoperability, efficiency, cost-benefit assessment, and security), as shown in Figure 7. These architectures demonstrate a robust capability for interoperability with legacy systems, ensuring seamless integration without necessitating extensive overhauls. This capability enables a gradual phase-out or complete overhaul of legacy systems without significant disruptions. However, the integration of SDN/NFV-based architectures can be challenging due to the legacy infrastructure commonly found in industrial environments, which may not be compatible with SDN protocols and architectures. Novel designs must ensure that the new SDN/NFV systems can seamlessly integrate with the existing network of 250 Ethernet switches and various NFV instances, thus maintaining continuity and operational efficiency.

Further, they facilitate the customization and configuration of advanced traffic engineering methodologies, which enhance network efficiency and ensure adherence to stipulated QoS requirements. Efficiency is highlighted as a major advantage, with SDN enabling path optimization to reduce latency and increase data transmission speed. In the context of a wind farm, this translates to real-time and rapid response time applications, which are essential for managing the vast amounts of data generated by 111 wind turbines.

Enhanced efficiency ensures that the network can handle high data loads more effectively, improving the overall operational performance.

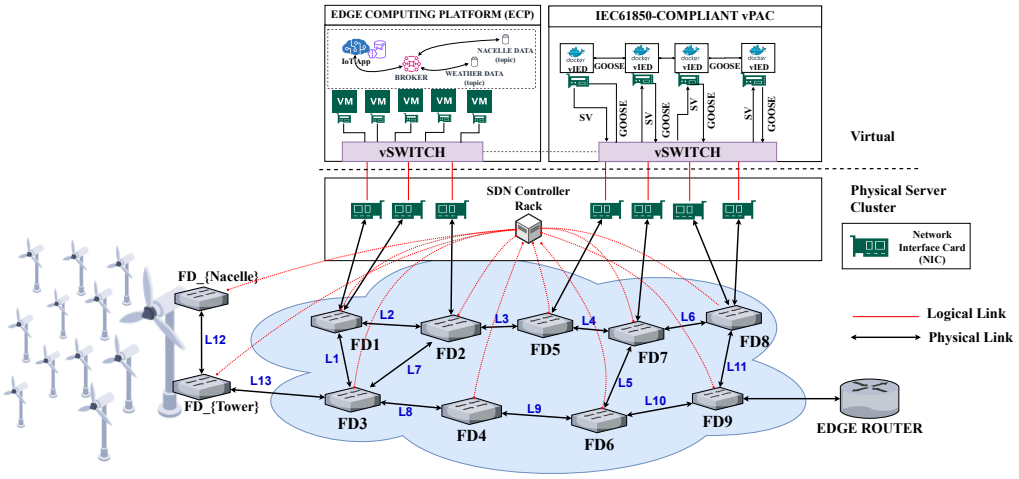


**Figure 6.** Software-defined IIoT–Edge network for next-generation offshore wind farm data-acquisition system (see Figure 3) connecting wind turbine generator access network (nacelle and tower access switches) to the pico data center ethernet switches.

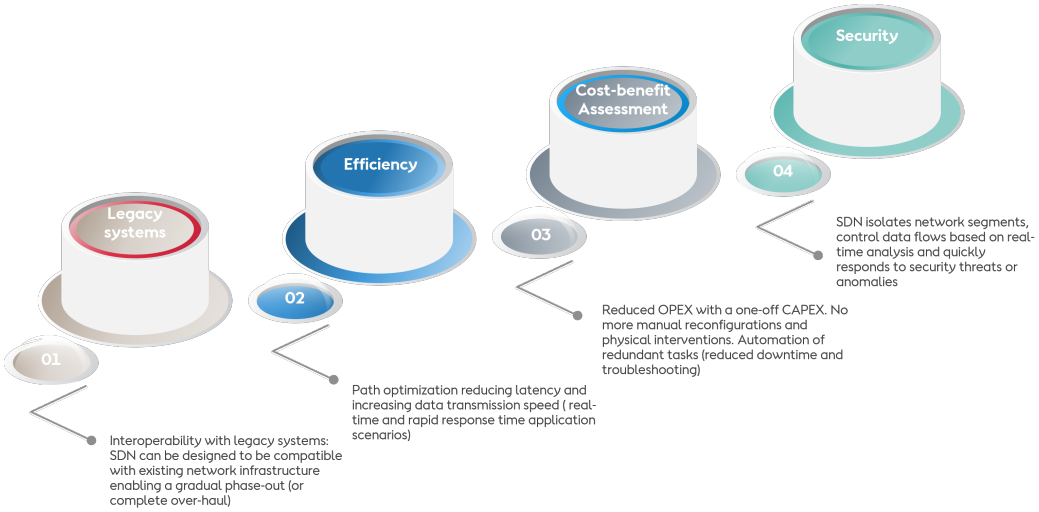


**Figure 7.** Considerations for adopting software-defined networking for industrial OT networks.

The SDN/NFV solutions are recognized for their cost-effectiveness, offering a financially viable alternative to traditional network management approaches. SDN/NFV adoption leads to reduced operational expenditure (OPEX) with one-off capital expenditure (CAPEX). This reduction in costs comes from minimizing manual reconfigurations and physical interventions, which are often time-consuming and costly. The automation of redundant tasks, such as those managed by the PDC in the wind farm, further reduces downtime and troubleshooting efforts, leading to significant cost savings and increased productivity.

Lastly, these architectures allow for the implementation of specialized security policies, thereby maintaining the integrity and security of the network. Security is a critical benefit of SDN, as it isolates network segments and controls data flows based on real-time analysis. This capability allows the network to quickly respond to security threats or anomalies. For industrial OT networks, such as the described wind farm, this means that sensitive data and critical infrastructure are better protected against cyber-attacks. The deployment of NFV instances within the PDC and enhances the overall security posture, ensuring that robust defense mechanisms are in place. Collectively, these attributes under-

score the potential of SDN/NFV-based architectures to significantly optimize industrial OT networks.

SDN/NFV-based architectures are still being researched, with most studies reviewing their interdependence and potential application scenarios. Several SDN/NFV-based architectures are designed for cyber-physical energy systems. Gjermundrod et al. [110] propose a "Gridstat" communication middleware framework based on a publisher–subscriber model to address the wide-area network-level limitations and achieve the timely and robust delivery of updates. Sakic et al. [21] proposed a middleware communication framework that deploys a software-defined network/network function virtualization (SDN/NFV)-based framework for the inter- and intra-domain management of wind park communication networks in the EU H2020 project "VirtuWind". Zopellaro et al. [111] reviewed SDN controllers for SDN protection and control systems, describing the role that SDN plays in transmitting delay-sensitive electrical messages such as the IEC 61850 GOOSE and SV in digital substations. Al Mhdawi et al. [19] designed a micro cloud-software-defined network testbed for onshore wind farm network recovery.

## 4. Performance, Security, Reliability, and Scalability Challenges of Software-Defined IIoT–Edge Networks

This section addresses the performance, security, reliability, and scalability challenges that hinder the implementation of software-defined IIoT–Edge networks in offshore wind farms' data acquisition systems.

### *4.1. Performance Concerns*

There is a need to model highly performing communication networks considering the criticality of offshore wind farm operations being dependent on data flowing through the data-acquisition system. The software-defined IIoT–Edge communication network performance is evaluated using metrics such as throughput, which measures the amount of data successfully transferred from source to destination; latency, which measures the time a packet travels from source to destination; packet loss, which measures the percentage of packets that are sent but do not reach their destination; bandwidth utilization, which measures how efficiently the network's bandwidth capacity is used; and QoS compliance, which evaluates how well the network supports the differentiated handling of packets to meet the requirements of the wind farm operator [100,112].

A highly performing network maximizes the throughput, QoS compliance, and bandwidth utilization and minimizes the latency and packet loss [113–115]. These software-defined IIoT–Edge communication networks must meet the industry SLAs as denoted in Table 1. To enhance performance, the implementation of redundancy strategies and failover mechanisms is essential to handle potential network failures without disrupting the wind farm operations. This involves deploying multiple SDN controllers and links to ensure that there is no single point of failure, thus maintaining continuous service availability and performance integrity.

Several standards are explored to assess the performance of such networks in offshore wind farms and other smart grids. Wang et al. [116] assess the end-to-end communication delay between communication devices in machine-to-machine communication application scenarios highlighting standards such as IEEE 1646 (communication delivery times within and external to an electrical substation), IEEE C37.1 (for wide-area situational awareness network performance in SCADA and automation systems), IEEE 1379 (communication protocol stack mapping for substation network functionality), and IEC 61850-5 (application recovery delay for substation bus communication) [65].

### *4.2. Security Concerns*

Adopting software-defined IIoT–Edge networks exposes the critical wind farm infrastructure to cyber threats. Cyber threats (a potential risk of exploiting a vulnerability) and attacks (an act of exploiting a vulnerability) can either be passive or active depending on

the objective of the intruder [117–119]. Cyber-attacks (i) allow the intruder to gain unauthorized access to data (*passive attack*) [120], (ii) modify data (such as electricity market data by adjusting dynamic prices, energy consumption data, vPAC and ECP node data, etc.) (*active attack*) [121], and (iii) trigger a loss of control over the energy system through the distributed denial of service (DDoS) attack (*active attack*) [122]. The intruders who develop methods to exploit vulnerabilities in this communication network are motivated by (i) financial, (ii) espionage, (iii) disruption, (iv) political, and (v) retaliation reasons [120,123,124]. In the last decade, several significant cyber incidents (in Europe and worldwide) have resulted in financial losses to the tune of millions of euros and a loss of electricity services to consumers for prolonged periods [125]. For example in April 2022, a large-scale communication network serving nearly 2000 wind turbines in Germany was targeted and compromised, paralyzing wind farm operations [126].

Intruders mainly gain access to industrial OT networks through the wide area network or the point of connection to the IT systems. These intruders collect sensitive information through phishing attacks, corrupted hyperlinks, and email attachments [126] and create malicious payload in Figure 8's ISA/IEC62443 Industrial Control System cyber kill chain Reconnaissance (planning) and Weaponization (preparation) stages [127,128]. For instance, intruders may gain unauthorized access to data at data centers or during transit, such as in man-in-the-middle (MiM) attacks. While some attackers only access data without altering them, others may modify data both in motion and at rest. They can disrupt the management plane and switch access to the SDN controller by flooding it with requests, depleting its memory and computing resources. Attackers may run recursive scripts on the controller to overwhelm its resources. By studying the TTL of flow entries on switches, attackers can divert traffic at the data plane. They may infiltrate switches to change flows, poison caches, and delete records. By impersonating authorized users or resources, attackers can disguise data packets, making it difficult to distinguish original data from counterfeit packets. Additionally, they may access applications hosting the communication framework to launch denial-of-service attacks or modify operations.

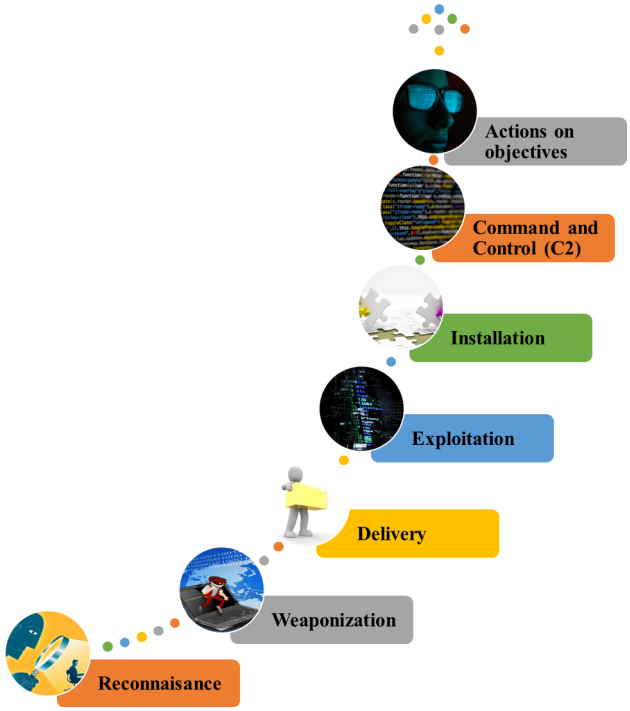


**Figure 8.** ISA/IEC62443 Industrial Control System Cyber-kill chain adapted from [126,127].

Next, these intruders deliver the malicious payload into the OT network and exploit vulnerabilities in the Delivery and Exploitation stages. Thereafter, they run the malicious software (mostly in the background to avoid detection) to collect data on the wind farm

operations in the Installation stage. Next, the intruders take control of the offshore wind farm data-acquisition system and cause damage by sending corrupted data to protection, automation, and control units in the Command and Control (C2) stage. Ultimately, this leads to partial or complete wind farm downtime and denies remote access to the O&M personnel.

To address the rising cybersecurity concern in this software-defined IIoT–Edge network, robust security policies, rules, and intrusion-detection models are adopted in both DIM and DAR modules. Presekal et al. [122] develop an attack graph model for cyber-physical power systems using hybrid deep learning to mitigate cyber threats on a digital substation level. Further, Mohan et al. [129] discuss the impact of a DDoS attack on the communication network for a load frequency control case scenario in power systems. From these two studies, novel attack models are developed to deal with cyber threats directed to the network domain (see data plane in Figure 5) of the IoT platforms in smart grids. Other vulnerability points besides the network domain are software domain and access domain vulnerabilities in the control and application plane [130].

### 4.3. Reliability Concerns

Like traditional communication networks, the software-defined IIoT–Edge network encounters stochastic disruptions, resulting in intermittent network service interruptions [131,132]. The reliability of the software-defined IIoT–Edge network addresses architectural robustness, failover mechanisms, and performance under stress conditions. Given that these networks are deployed in extreme environments with unpredictable weather, there is a need to deploy ruggedized equipment that can withstand the harsh conditions of the offshore environment.

Failures can occur at various levels, including the data plane (like switches and routers), the control plane (controller software or the applications running on it), and the data links connecting different network elements [100]. The impact of such failures is magnified in offshore wind farms due to their remote locations, harsh operating conditions, and the critical nature of the operations they support, which include safety systems, power generation monitoring, and environmental controls. Fast failover mechanisms are crucial in SDN to ensure that the network services remain uninterrupted during and after failures. SDN controllers can pre-compute alternate paths in the network, which can be quickly activated in the event of a failure [133]. This proactive approach significantly reduces the recovery time compared to traditional methods. Several reliability metrics are used to evaluate how well a network performs over time [134,135], such as Failure rate, which measures the frequency with which a network component or service fails; Availability, which measures the percentage of time that the network is operational and available for use; Mean time between failures (MTBF), which measures the average time between inherent system failures during operation; and Mean time to repair (MTTR), which measures the average time required to repair a failed network component and restore it to operational status.

Industrial communication networks adhere to industrial SLAs. For example, the desired communication network availability is 99.999% (*5 nines*), which implies only 5 min of downtime in a year [136]. In practice, the network performance and availability are constantly affected, and, as a result, more advanced fast failover recovery methods are studied to tackle the availability issues by defining reliability or failure models and making inferences from the model results to determine the availability [137].

### 4.4. Scalability Concerns

The SDN controller is often viewed as a central point of failure for the software-defined IIoT–Edge network. *n* SDN controllers are deployed to create redundancy and improve flow initialization computation rates. These SDN controllers are connected through East–West interfaces to form a cluster, as illustrated in Figure 5. In practice, the SDN controller software instances are shipped as *karaf* containers. This software instance runs on a virtual machine with sufficient CPU and memory resources. These CPU and memory resources

must be constantly monitored to ensure that they do not exceed a stipulated threshold above which the performance and overall health of the SDN controller are degraded [48].

Furthermore, the SDN controller placement problem (CPP) has been of critical concern, especially in designing large-scale networks where deploying a single network is not feasible [95]. Studies solve the CPP by modeling approaches that deploy the optimal number of controllers within a network while meeting certain performance requirements. Studies by Saleh et al. [138], Lu et al. [139], Aly et al. [140], and Singh et al. [141] explore heuristic and optimization algorithms to determine the ideal number of SDN controllers for a given network size and the optimal placement in proximity to the FDs and other SDN controllers.

Furthermore, the SDN controller must manage communication with a potentially vast number of switches and other network devices, which can become a bottleneck as the network grows. Scalability concerns also extend to the software layers, where the controller's ability to process and respond to network events in real time is vital. To address this, SDN architectures may employ distributed controllers, hierarchical designs, or clustering techniques to enhance scalability [142]. On the positive side, the programmability of SDN enables dynamic resource allocation and network adjustments, allowing for elastic scalability that can accommodate variable workloads and changing network topologies [143].

## 5. Building Resilience in Software-Defined IIoT–Edge Networks

This section reviews the resilience approaches and the self-X autonomous network management framework proposed in the literature to build resilience and adapt these approaches to suit software-defined IIoT–Edge networks.

### *5.1. Resilience in Software-Defined IIoT–Edge Networks*

Madni et al. [144] developed a conceptual system recovery curve that illustrates the system's performance before disruption and during disruption and the gradual rise to recovery, as illustrated in Figure 9. The system performance is denoted as, $y_{(t)}$, with respect to the operational time, $[0, t)$. Before the disruption, the system performs optimally at $y_{(t)} = y_{(rf)}$. At the time $t_D$, the system encounters disruptions that lower the performance to the minimum acceptable levels, $y_m$. At this point, recovery mechanisms and strategies are initiated to help restore the system to full recovery ($y_{(rf)}$) or partial recovery ($y_{(rp)}$) at the time $t_R$. Lastly, the time between disruption and full or partial recovery is given as $\Delta t = (t_R - t_D)$. Averaging samples of $\Delta t$ over time yields the mean time to recovery (MTTR). Poulin et al. [131] advanced this resilience curve using normalized performance metrics and the well-known Weibull distribution bath curve to demonstrate disruption and recovery.

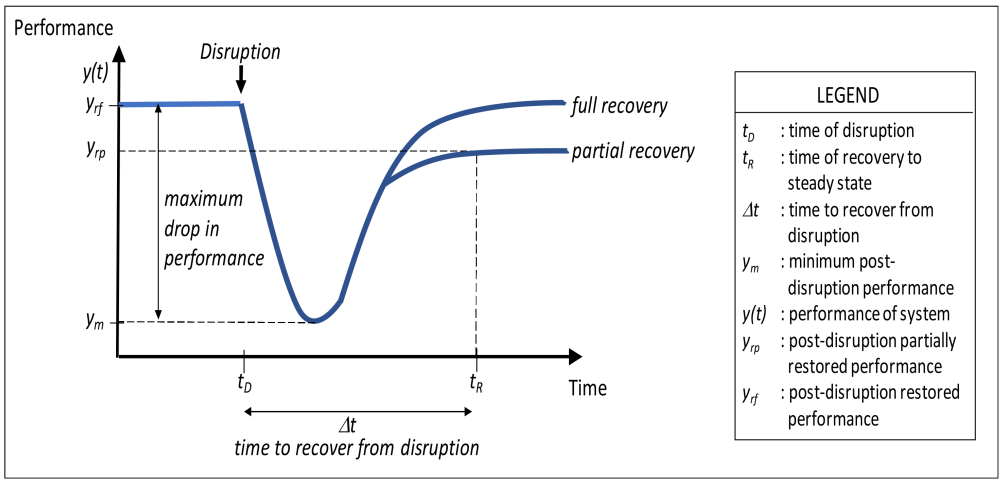


**Figure 9.** An illustration of a system's capacity to recover from disruptions within time [0, t) [144].

To build resilience in the system recovery curve specifically for a communication network, two conditions must be met: (i) the performance must not reach or exceed the maximum drop in performance point, $y_{(m)}$, to meet the industrial SLAs, and (ii) the time to recover from disruption, $\Delta t$, must not exceed the stipulated industrial MTTR.

To meet this objective, several approaches are taken to address the disruptions, minimizing the drop in performance and the time taken to recover from the disruption to improve overall resilience:

**Redundancy and load balancing:** The first approach is to build redundancy in the communication network. This ensures that the failure of a controller or forwarding device does not interrupt network service. Further, with data links, the EtherChannel link aggregation technique is used in this network [145]. The technology bundles together several physical cables running between two forwarding devices into a single logical link. Lastly, using load balancers (as physical or network function virtualization instances) enables the even distribution of traffic across multiple paths to prevent congestion and single points of failure [146].

**Fault detection, failover, and recovery:** Application-plane-customized RestAPI applications, alongside flow monitoring tools such as NetFlow, sFlow, or IPFIX, are used to collect flow statistics, analyze traffic patterns, and detect anomalies indicative of faults. These applications monitor device responsiveness, detecting unreachable devices or those performing suboptimally. Fast failure recovery within a fixed time interval is needed to support network services [70]. Gyllstrom et al. [147] designed a link failure detection and reporting mechanism that uses OpenFlow to detect link failures within the data plane network topology. To facilitate failover and recovery, [147] formulated the *"MULTICAST RECYCLING"* algorithm that precomputes backup multicast trees to recover from link failure. Further, Petale et al. [148] proposed a new scheme, the Group Table Rerouting (GTR) technique, to find the response against single link failure through the fast fail-over (FF) group table feature provided by OpenFlow. Lastly, Miura et al. [149] introduced a fast failure recovery mechanism based on multiple routing configuration algorithms using programming protocol-independent packet processors (P4).

**Traffic engineering and QoS policies:** The SDN controller, with a global view of the entire network, analyzes the network traffic patterns and makes decisions about how to route traffic based on real-time conditions [100]. In practice, the network engineer defines the traffic engineering policies based on the industrial SLAs or business objectives (*"intents"*) for traffic prioritization or path optimization. These *intents* are translated into flow rules that the SDN controller can implement in the network. Additionally, QoS policies specify resource allocation, for example, bandwidth allocation, latency requirements, packet loss tolerance, etc., for different types of traffic or applications. QoS-related policies are configured in the SDN controller to enforce policies by applying the appropriate QoS treatment for each QoS class [150]. Guo et al. [151] developed a reinforcement learning (RL) traffic engineering method that trains a traffic-splitting RL agent to address the dynamically changing traffic and achieve link load balancing. Further, Keshari et al. [152] highlighted the implementation of QoS features in different open-source SDN controllers such as ONOS (Java-based *SetQueueInstruction* functionality), ODL-Lithium (Southbound plugin for the DOCSIS infrastructure), and the Floodlight Northbound interface (which runs the module *QueuePusher*), which generates messages for queue configurations to Create, Read, Update, and Delete flows in the openflow-enabled switches flow tables.

**Network isolation and segmentation:** This highlights crucial techniques adopted to divide the network into multiple segments or zones defining security and traffic engineering policies for each zone [153]. It reduces the attack surface because attackers may gain access to one zone and not the other zones, making it significantly harder to attack the entire network. This contains the security breach and prevents it from spreading to other areas of the communication network. Additionally, the network performance is improved by limiting broadcast traffic to specific segments and enhancing fault tolerance by isolating failures [154]. The SDN controller defines the network segments using virtual local area

networks (VLANs) to segregate the network traffic. Using RESTAPI-based applications running in the application plane, the SDN controller can implement access control lists (ACLs) and flow rules that enforce the isolation between segments.

### *5.2. Autonomous Networks: Self-X Network Management*

Given the growing complexity of offshore wind farm networks, strict security and performance demands, and the difficult accessibility of the offshore environment, it is essential for network engineers to adopt automation and intelligence for network and service management. This shift from manual, reactive resilience measures to proactive, automated strategies will significantly improve operational resilience, performance, and scalability.

According to the ETSI ZSM standard group, *"An autonomous network (AN) is a network that self-operates according to the business goals with no human intervention beyond the initial supply of input (e.g., intent, goals, policies, certain configuration data) by the human operator"* [155,156]. The ultimate goal in network automation is to develop networks that can operate independently, guided by high-level policies set by the provider or user. These autonomous networks can configure, monitor, heal, and optimize themselves without human input [157–159]. Achieving this necessitates the creation of a new architectural framework that supports closed-loop control and is tailored for applying data-driven AI algorithms [160]. These autonomous networks define self-X network and service management properties such as self-configuration, "this property relates to the autonomic capability of the AN to configure and govern any parameters or settings which relate to functions, services, or assets that make up or relate to that autonomous network"; self-healing, and self-repair, *AN's ability to fix anomalies after detecting them and to get back to normal fully operational mode through a process of automated steps"*; and self-awareness, *"cognitive-like awareness regarding several dimensions such as context information, time, SLAs, KPIs, environments"* [23,161,162].

Figure 10 illustrates an autonomous software-defined IIoT–Edge network for an offshore wind farm application. In the design and implementation of the network, high-level policies such as specific performance metrics, security policies, and traffic engineering rules are defined as business goals [157,163]. These business goals, formulated by the O&M team, are translated into intents that guide the autonomous operation of the network using an abstraction module [158]. The network implements a monitoring feature at the application plane to collect data on network performance traffic flows, device status, and link status. The network uses a custom-based machine learning (ML) model to recommend configuration changes or take corrective actions in the *"observe-orient-decide-act"* cycle (see the application plane in Figure 10) [164–167].

In this *"observe-orient-decide-act"* cycle, the network-monitoring module reads the network state and stores it in a knowledge base (such as Redis or Hadoop). Further, it forwards the network state to the predictive analytics module for visualization and forecasting purposes. The recommender module accesses the network state stored in the knowledge base, which formulates traffic engineering functions to manage resources on the network, guaranteeing network performance and availability.

To make the network fully autonomous, the AI/ML decision-maker module uses model-based and model-free reinforcement learning approaches to assist the recommender module in optimizing traffic routing and anomaly detection to respond to security threats or failures. Further, the AI/ML decision-maker module updates the knowledge base with data for future use [168,169]. These modules provide feedback to the network administrators on the impact of the high-level policies on the network, enabling them to decide how to effect the changes.

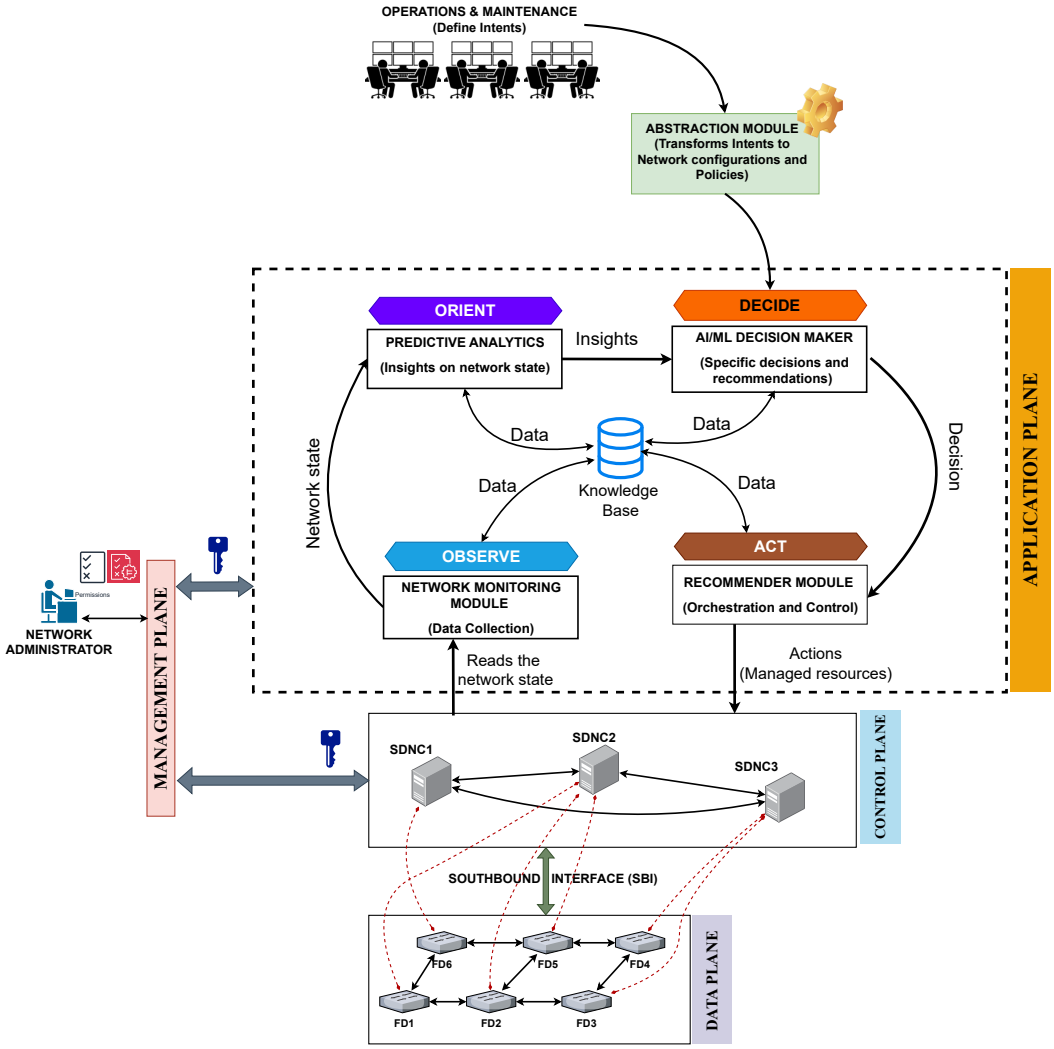


**Figure 10.** A self-managing software-defined IIoT–Edge network tailored for next-generation offshore wind farms (see Figure 6) by incorporating the European Telecommunication Standards Institute (ETSI) zero-touch network and service management framework [156].

Several studies have explored reinforcement learning approaches to realize autonomous software-defined networks across different use cases. These RL strategies deploy single or multi-agents at the application plane to interact directly with the system, learn, and create optimal policies to meet the stipulated intents. For example, Dake et al. [170] developed a *"multi-agent deep deterministic policy gradient (MA-DDPG)"* framework that facilitates transient load (traffic burst, elephant flow, and mice flow) detection and prevention in SDN-IoT networks. The framework uses two RL agents, where one agent ensures efficient multipath routing optimization while the other agent ensures malicious DDoS traffic detection and prevention. Passito et al. [171] developed *"AgNOS,"* an agent-based framework for the autonomous control of software-defined networks to address the arduous task of handling the total distribution of control between autonomous systems. The framework uses agents to mitigate DDoS attacks on several autonomous systems. Further, Yao et al. [172] designed *"NetworkAI"*, an intelligent network architecture for self-learning control strategies in software-defined networks. Hu et al. [173] developed *"EARS"*, an intelligent, experiential network architecture utilizing deep reinforcement learning (DRL) for autonomous routing in software-defined networks. The DRL strategy integrates a closed-loop control system with network monitoring technologies to manage network environments plagued by link congestion and inefficient bandwidth distribution across flows. Lastly, Casas-Velasco et al. [174] proposed *"Reinforcement Learning and Software Defined Networking for*

*Intelligent Routing (RSIR)"*, a novel approach to SDN routing that uses a knowledge plane to monitor the network, gather link-state data, and identify optimal paths for intelligent routing amidst traffic fluctuations. While still a subject of ongoing research, the frameworks proposed by [170–174] incur significant computational costs during training, complicating the creation of an optimal AI/ML decision-maker module. Additionally, security is a major concern in developing and deploying autonomous systems. Managing the extensive data and network parameters knowledge base also presents challenges, particularly due to the rapid rate at which the network monitoring module samples data.

Leveraging the power of AI and ML, these frameworks can autonomously manage and optimize network traffic and proactively address the performance, security, reliability, and scalability challenges before they impact network operations. Integrating such technologies into SDNs transforms traditional networking infrastructures into intelligent, self-optimizing systems that significantly enhance operational efficiency and reliability. This paradigm shift toward intelligent networks is crucial for handling the increasing complexity and scale of modern network demands, ultimately paving the way for more resilient networking architectures.

## 6. Limitations of the Study

While this paper provides a comprehensive review of IIoT–Edge networks and their potential for transforming next-generation offshore wind farms, there are several limitations to note. The study does not delve into the ongoing standardization efforts aimed at readying SDN/NFV solutions for widespread adoption in offshore wind farms and other industrial OT networks. Furthermore, it does not consider the change-management requirements critical for adopting these innovative technologies. Resistance to change and the need for extensive testing and validation to ensure the reliability and safety of OT systems for mission-critical applications are significant factors that could slow down the adoption process. Addressing these areas would offer a more holistic understanding of the challenges and readiness of SDN/NFV technologies in industrial applications.

## 7. Conclusions and Future Work

This review paper has explored the state of the art in IIoT–Edge networks for high-availability, consistent-performance-demanding environments like the next-generation offshore wind farms, or "wind farms of tomorrow". The paper illustrated how Industry 4.0 technologies such as IIoT, Edge computing, and virtualization are integrated into the design of offshore wind farm data acquisition systems. The paper, then examined key protocols involved in transmitting data within these systems that support offshore wind farm operations. Although traditional communication networks could facilitate data transfer in next-generation offshore wind farms, they come with significant drawbacks: high deployment costs, complex management at scale, difficulty in configuration and maintenance due to their distributed nature, and vendor lock-in that hampers innovation and customization.

This paper discussed two pivotal technologies—SDN (software-defined networking) and NFV (network function virtualization)—that will revolutionize IIoT–Edge data-acquisition systems by making networks more software-defined, programmable, and vendor-neutral. Unlike traditional networks, SDN/NFV-based networks facilitate dynamic, centralized control, allowing network administrators to rapidly diagnose and rectify faults, as well as to adapt the network to meet specific business intents, objectives, or industrial service-level agreements. Moreover, these networks boast significantly faster convergence times, achieving rates of 100 µs, compared to the 10–30 ms typical of traditional networks.

Despite their potential, software-defined IIoT–Edge networks face performance, reliability, security, and scalability challenges that may impede their implementation. This review has addressed these challenges and highlighted various mitigation strategies such as redundancy and load balancing, fault detection and recovery, traffic engineering with Quality of Service (QoS) policies, network isolation, and segmentation. Finally, the paper

reviewed AI-driven self-X (self-configuring, self-healing, and self-optimizing) approaches that not only autonomously manage and optimize network traffic but also proactively tackle performance, security, reliability, and scalability challenges before they affect operations. This development is increasingly vital for addressing the complex and growing demands of software-defined IIoT–Edge networks, suggesting a pivotal direction for future research in creating more resilient networking architectures. These strategies are vital for fostering robust, reliable, and high-performing software-defined IIoT–Edge networks that can fully support the sophisticated requirements of next-generation offshore wind farms.

As a main advantage, software-defined networks provide dynamic reconfiguration capabilities, allowing the communication network to adapt to changing conditions and demands in real time. Such flexibility will have a crucial role in supporting the anticipated developments in offshore renewable energy systems, consisting of (multiple) large-scale wind farms and other renewable energy generation technologies (e.g., wave and floating solar photovoltaic power), offshore electrolyzers, electricity and hydrogen storage systems, etc. The deployment of software-defined IIoT–Edge networks to facilitate a more resilient and adaptable cyber-physical energy system that includes heterogeneous energy conversion, storage, and transport technologies in a harsh offshore environment represents a promising future research direction.

**Author Contributions:** Conceptualization, A.M.; Methodology, A.M., E.F., M.G. (Madeleine Gibescu); Resources, R.S., M.G. (Mikkel Gryning); Writing—original draft preparation, A.M.; Writing—review and editing, A.M., E.F., R.S., M.G. (Mikkel Gryning), M.G. (Madeleine Gibescu); Visualization, A.M.; Supervision, E.F., M.G. (Mikkel Gryning), M.G. (Madeleine Gibescu); Funding acquisition, M.G. (Madeleine Gibescu). All authors have read and agreed to the published version of the manuscript.

**Funding:** This research is part of the Innovative Tools for Cyber-Physical Energy Systems (InnoCyPES) project. The project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska Curie grant agreement No 956433.

**Data Availability Statement:** No new data were created or analyzed in this study.

**Conflicts of Interest:** Author Mikkel Gryning was employed by the company Ørsted Wind Power at the time of publication. The views expressed in this paper are solely his own personal views and should not be taken to represent the views of the company. All authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

| | |
|---|---|
| AMQP | Advanced Message Queueing Protocol |
| COAP | Constrained Application Protocol |
| DAR | Data-at-Rest |
| DIM | Data-in-Motion |
| ECP | Edge Computing Platform |
| ETSI | European Telecommunication Standards Institute |
| FD | Forwarding Device |
| GOOSE | Generic Object-Oriented Substation Events |
| IEC | The International Electrotechnical Commission |
| IIoT | Industrial Internet of Things |
| Industry 4.0 | Fourth Industrial Revolution |
| IT/OT | Information Technology/Operational Technology |
| ISA | International Society of Automation |
| KPI | Key Performance Indicators |
| LCOE | Levelized Cost of Energy |
| MMS | Manufacturing Message Specification |
| MQTT | Message Query Telemetry Transport |
| MU | Merging Unit |

| | |
|---|---|
| NFV | Network Function Virtualization |
| O&M | Operations and Maintenance |
| OPC-UA | Open Platform Communication Unified Architecture |
| PDC | Pico-Data Center |
| QoS | Quality of Service |
| RESTful API | Representational State Transfer Application Programming Interface |
| SDN | Software Defined Networking |
| SLA | Service Level Agreement |
| SQL | Structured Query Language |
| vIED | virtual Intelligent Electronic Device |
| vPAC | virtual Protection, Automation, and Control |

## References


1. Ren, Z.; Verma, A.S.; Li, Y.; Teuwen, J.J.; Jiang, Z. Offshore wind turbine operations and maintenance: A state-of-the-art review. *Renew. Sustain. Energy Rev.* **2021**, *144*, 110886. [CrossRef]
2. Cao, X.; Xu, Y.; Wu, Z.; Qin, X.; Ye, F. Data acquisition and management of wind farm using edge computing. *Int. J. Grid Util. Comput.* **2022**, *13*, 249–255. [CrossRef]
3. Xiang, X.; Gui, J.; Xiong, N.N. An integral data gathering framework for supervisory control and data acquisition systems in green IoT. *IEEE Trans. Green Commun. Netw.* **2021**, *5*, 714–726. [CrossRef]
4. Swiszcz, G.; Cruden, A.; Booth, C.; Leithead, W. A data acquisition platform for the development of a wind turbine condition monitoring system. In Proceedings of the 2008 International Conference on Condition Monitoring and Diagnosis, Beijing, China, 21–24 April 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 1358–1361.
5. Kou, L.; Li, Y.; Zhang, F.; Gong, X.; Hu, Y.; Yuan, Q.; Ke, W. Review on monitoring, operation, and maintenance of smart offshore wind farms. *Sensors* **2022**, *22*, 2822. [CrossRef] [PubMed]
6. Rinaldi, G.; Thies, P.R.; Johanning, L. Current status and future trends in the operation and maintenance of offshore wind turbines: A review. *Energies* **2021**, *14*, 2484. [CrossRef]
7. Tightiz, L.; Yang, H. A comprehensive review on IoT protocols' features in smart grid communication. *Energies* **2020**, *13*, 2762. [CrossRef]
8. Sikarwar, R.; Yadav, P.; Dubey, A. A Survey on IOT enabled cloud platforms. In Proceedings of the 2020 IEEE 9th International Conference on Communication Systems and Network Technologies (CSNT), Gwalior, India, 10–12 April 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 120–124.
9. Zhang, P.; He, Z.; Cui, C.; Xu, C.; Ren, L. An edge-computing framework for operational modal analysis of offshore wind-turbine tower. *Ocean Eng.* **2023**, *287*, 115720. [CrossRef]
10. Xu, Y.; Nascimento, N.M.M.; de Sousa, P.H.F.; Nogueira, F.G.; Torrico, B.C.; Han, T.; Jia, C.; Reboucas Filho, P.P. Multi-sensor edge computing architecture for identification of failures short-circuits in wind turbine generators. *Appl. Soft Comput.* **2021**, *101*, 107053. [CrossRef]
11. Torres, E.; Eguia, P.; Abarrategi, O.; Larruskain, D.; Valverde, V.; Buigues, G. Trends in Centralized Protection and Control in Digital Substations. *RE&PQJ* **2023**, *21*, 196–201. [CrossRef]
12. Kabbara, N.; Mwangi, A.; Gibescu, M.; Abedi, A.; Stefanov, A.; Palensky, P. Specifications of a Simulation Framework for Virtualized Intelligent Electronic Devices in Smart Grids Covering Networking and Security Requirements. In Proceedings of the 2023 IEEE Belgrade PowerTech, Belgrade, Serbia, 25–29 June 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1–6.
13. Netes, V. New international standard for dependability. *Dependability* **2016**, *3*, 54–58. [CrossRef]
14. Netes, V. Dependability measures for access networks and their evaluation. In Proceedings of the 2020 26th Conference of Open Innovations Association (FRUCT), Yaroslavl, Russia, 20–24 August 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 352–358.
15. Misra, K.B. *Handbook of Performability Engineering*; Springer Science & Business Media: London, UK, 2008.
16. Jheeta, M.S. Resilience, Reliability, and Recoverability (3Rs). Master's Thesis, UiT The Arctic University of Norway, Tromsø, Norway, 2022.
17. Dorsch, N.; Kurtz, F.; Georg, H.; Hagerling, C.; Wietfeld, C. Software-defined networking for smart grid communications: Applications, challenges and advantages. In Proceedings of the 2014 IEEE International Conference on Smart Grid Communications (SmartGridComm), Venice, Italy, 3–6 November 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 422–427.
18. Doherty, J. *SDN and NFV Simplified: A Visual Guide to Understanding Software Defined Networks and Network Function Virtualization*; Addison-Wesley Professional: Boston, MA, USA, 2016.
19. Al Mhdawi, A.K.; Al-Raweshidy, H. mSDN: Micro Cloud-Software Defined Network Testbed for Onshore Wind Farm Network Recovery. In Proceedings of the 2018 IEEE Global Conference on Internet of Things (GCIoT), Alexandria, Egypt, 5–6 December 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1–6.
20. Vizarreta, P.; Van Bemten, A.; Sakic, E.; Abbasi, K.; Petroulakis, N.E.; Kellerer, W.; Machuca, C.M. Incentives for a softwarization of wind park communication networks. *IEEE Commun. Mag.* **2019**, *57*, 138–144. [CrossRef]

21. Sakic, E.; Kulkarni, V.; Theodorou, V.; Matsiuk, A.; Kuenzer, S.; Petroulakis, N.E.; Fysarakis, K. VirtuWind–An SDN-and NFV-based architecture for softwarized industrial networks. In Proceedings of the International Conference on Measurement, Modelling and Evaluation of Computing Systems, Erlangen, Germany, 26–28 February 2018; Springer: Cham, Switzerland, 2018; pp. 251–261.
22. Leivadeas, A.; Falkner, M. A survey on intent-based networking. *IEEE Commun. Surv. Tutor.* **2022**, *25*, 625–655. [CrossRef]
23. Liyanage, M.; Pham, Q.V.; Dev, K.; Bhattacharya, S.; Maddikunta, P.K.R.; Gadekallu, T.R.; Yenduri, G. A survey on Zero touch network and Service Management (ZSM) for 5G and beyond networks. *J. Netw. Comput. Appl.* **2022**, *203*, 103362. [CrossRef]
24. Mwangi, A.; Sundsgaard, K.; Leiva Vilaplana, J.A.; Vilerá, K.V.; Yang, G. A System-Based Framework for Optimal Sensor Placement in Smart Grids. In Proceedings of the 2023 IEEE Belgrade PowerTech, Belgrade, Serbia, 25–29 June 2023; pp. 1–6. [CrossRef]
25. Mustafa, A.; Markeset, T.; Barabadi, A. Downtime cost estimation: A wind farm in the arctic case study. In Proceedings of the 30th European Safety and Reliability Conference and the 15th Probabilistic Safety Assessment and Management Conference (ESREL), Venice, Italy, 1–5 November 2020.
26. Wiser, R.; Bolinger, M.; Lantz, E. Assessing wind power operating costs in the United States: Results from a survey of wind industry experts. *Renew. Energy Focus* **2019**, *30*, 46–57. [CrossRef]
27. *IEC 61400-25*; International Standard on Communications for Monitoring and Control of Wind Power Plants, TC 88—Wind Energy Generation Systems. International Electrotechnical Commission (IEC): Geneva, Switzerland, 2017.
28. Wang, H.; Xiong, B.; Zhang, Z.; Zhang, H.; Azam, A. Small wind turbines and their potential for Internet of things applications. *Iscience* **2023**, *26*. [CrossRef] [PubMed]
29. Walford, C.A. *Wind Turbine Reliability: Understanding and Minimizing Wind Turbine Operation and Maintenance Costs*; Technical Report; Sandia National Laboratories (SNL): Albuquerque, NM, USA; Livermore, CA, USA, 2006.
30. Karad, S.; Thakur, R. Efficient monitoring and control of wind energy conversion systems using Internet of things (IoT): A comprehensive review. *Environ. Dev. Sustain.* **2021**, *23*, 14197–14214. [CrossRef]
31. Sayed, K.; Abo-Khalil, A.G.; Eltamaly, A.M. Wind Power Plants Control Systems Based on SCADA System. In *Control and Operation of Grid-Connected Wind Energy Systems. Green Energy and Technology*; Springer: Cham, Switzerland, 2021; pp. 109–151.
32. Adekanbi, M.L. Optimization and digitization of wind farms using Internet of things: A review. *Int. J. Energy Res.* **2021**, *45*, 15832–15838. [CrossRef]
33. Zhou, F.; Tu, X.; Wang, Q. Research on offshore wind power system based on Internet of Things technology. *Int. J. Low-Carbon Technol.* **2022**, *17*, 645–650. [CrossRef]
34. El Hakim, A. Internet of Things (IoT) System Architecture and Technologies. White Paper. 2018. Available online: https://www.researchgate.net/publication/323525875_Internet_of_Things_IoT_System_Architecture_and_Technologies_White_Paper (accessed on 28 April 2024).
35. Tewolde, S.; Hoffer, R.; Haardt, H.; Krieger, J. Lessons learned from practical structural health monitoring of offshore wind turbine support structures in the North Sea. In Proceedings of the Final Conference of WINERCOST & Aeolus4future, Catanzaro, Italy, 21–23 March 2018; p. 153.
36. Loughney, S.; Edesess, A.J. Applications of industrial iot and wsns in o&m programmes for offshore wind farms. In *Computational Sciences and Artificial Intelligence in Industry*; Springer: Cham, Switzerland, 2022; pp. 223–245.
37. Minh, Q.N.; Nguyen, V.H.; Quy, V.K.; Ngoc, L.A.; Chehri, A.; Jeon, G. Edge Computing for IoT-Enabled Smart Grid: The Future of Energy. *Energies* **2022**, *15*, 6140. [CrossRef]
38. Cao, K.; Liu, Y.; Meng, G.; Sun, Q. An overview on edge computing research. *IEEE Access* **2020**, *8*, 85714–85728. [CrossRef]
39. Vijayarani, S.; Sharmila, S. Research in big data: An overview. *Inf. Eng. Int. J.* **2016**, *4*, 1–20.
40. Saeed, N.; Husamaldin, L. Big data characteristics (V's) in industry. *Iraqi J. Ind. Res.* **2021**, *8*, 1–9. [CrossRef]
41. Cappa, F.; Oriani, R.; Peruffo, E.; McCarthy, I. Big data for creating and capturing value in the digitized environment: Unpacking the effects of volume, variety, and veracity on firm performance. *J. Prod. Innov. Manag.* **2021**, *38*, 49–67. [CrossRef]
42. Yu, W.; Liang, F.; He, X.; Hatcher, W.G.; Lu, C.; Lin, J.; Yang, X. A survey on the edge computing for the Internet of Things. *IEEE Access* **2017**, *6*, 6900–6919. [CrossRef]
43. Atlam, H.F.; Alenezi, A.; Alharthi, A.; Walters, R.J.; Wills, G.B. Integration of cloud computing with Internet of things: Challenges and open issues. In Proceedings of the 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Exeter, UK, 21–23 June 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 670–675.
44. Ma, Y.; Zhao, F.; Zhou, X.; Gao, Z. Summary of cloud computing technology in smart grid. In Proceedings of the 2018 IEEE International Conference on Mechatronics and Automation (ICMA), Changchun, China , 5–8 August 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 253–258.
45. Yigit, M.; Gungor, V.C.; Baktir, S. Cloud computing for smart grid applications. *Comput. Netw.* **2014**, *70*, 312–329. [CrossRef]
46. Amoretti, M.; Pecori, R.; Protskaya, Y.; Veltri, L.; Zanichelli, F. A scalable and secure publish/subscribe-based framework for industrial IoT. *IEEE Trans. Ind. Inform.* **2020**, *17*, 3815–3825. [CrossRef]
47. Sultangazin, A.; Tabuada, P. Towards the use of symmetries to ensure privacy in control over the cloud. In Proceedings of the 2018 IEEE Conference on Decision and Control (CDC), Miami, FL, USA , 17–19 December 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 5008–5013.

48. Urrea, C.; Benítez, D. Software-defined networking solutions, architecture and controllers for the industrial Internet of things: A review. *Sensors* **2021**, *21*, 6585. [CrossRef] [PubMed]
49. Rafique, W.; Qi, L.; Yaqoob, I.; Imran, M.; Rasool, R.U.; Dou, W. Complementing IoT services through software-defined networking and edge computing: A comprehensive survey. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1761–1804. [CrossRef]
50. Mishra, B.; Kertesz, A. The use of MQTT in M2M and IoT systems: A survey. *IEEE Access* **2020**, *8*, 201071–201086. [CrossRef]
51. Elhadi, S.; Marzak, A.; Sael, N.; Merzouk, S. Comparative study of IoT protocols. In Proceedings of the Smart Application and Data Analysis for Smart Cities (SADASC'18), Casablanca, Moroco, 27–28 February 2018.
52. Moraes, T.; Nogueira, B.; Lira, V.; Tavares, E. Performance comparison of IoT communication protocols. In Proceedings of the 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), Bari, Italy , 6–9 October 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 3249–3254.
53. Silva, D.; Carvalho, L.I.; Soares, J.; Sofia, R.C. A performance analysis of Internet of things networking protocols: Evaluating MQTT, CoAP, OPC UA. *Appl. Sci.* **2021**, *11*, 4879. [CrossRef]
54. Bansal, M.; Priya. Performance comparison of MQTT and CoAP protocols in different simulation environments. In *Inventive Communication and Computational Technologies: Proceedings of ICICCT 2020*; Springer: Singapore, 2021; pp. 549–560.
55. Glaroudis, D.; Iossifides, A.; Chatzimisios, P. Survey, comparison and research challenges of IoT application protocols for smart farming. *Comput. Netw.* **2020**, *168*, 107037. [CrossRef]
56. Banno, R.; Shudo, K. Adaptive topology for scalability and immediacy in distributed publish/subscribe messaging. In Proceedings of the 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC), Madrid, Spain, 13–17 July 2020.
57. Al-Masri, E.; Kalyanam, K.R.; Batts, J.; Kim, J.; Singh, S.; Vo, T.; Yan, C. Investigating messaging protocols for the Internet of Things (IoT). *IEEE Access* **2020**, *8*, 94880–94911. [CrossRef]
58. Pu, C.; Ding, X.; Wang, P.; Yang, Y. Practical implementation of an OPC UA multi-server aggregation and management architecture for IIoT. In Proceedings of the 2022 IEEE International Conferences on Internet of Things (iThings) and IEEE Green Computing & Communications (GreenCom) and IEEE Cyber, Physical & Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics), Espoo, Finland, 22–25 August 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 476–481.
59. Peeters, C.; Daems, P.J.; Verstraeten, T.; NOWÃ, A.; Helsen, J. Combining Edge and Cloud Computing for Monitoring a Fleet of Wind Turbine Drivetrains Using Combined Machine Learning Signal Processing Approaches. In *Structural Health Monitoring 2019*; Vrije Universiteit Brussel: Brussel, Belgium, 2019.
60. Jun, H.J.; Yang, H.S. Performance of the XMPP and the MQTT protocols on IEC 61850-based micro grid communication architecture. *Energies* **2021**, *14*, 5024. [CrossRef]
61. Ferreira, R.D.F.; de Oliveira, R.S. Cloud IEC 61850 A Case Study of a Software Defined Protection, Automation & Control System. In Proceedings of the 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), Turin, Italy, 4–7 September 2018; IEEE: Piscataway, NJ, USA, 2018; Volume 1, pp. 75–82.
62. Rodríguez, M.; Lázaro, J.; Bidarte, U.; Jiménez, J.; Astarloa, A. A Fixed-Latency Architecture to Secure GOOSE and Sampled Value Messages in Substation Systems. *IEEE Access* **2021**, *9*, 51646–51658. [CrossRef]
63. Kim, J.; Filali, F.; Ko, Y.B. Trends and potentials of the smart grid infrastructure: From ICT sub-system to SDN-enabled smart grid architecture. *Appl. Sci.* **2015**, *5*, 706–727. [CrossRef]
64. Al-Ali, A.R.; Zualkernan, I.A.; Rashid, M.; Gupta, R.; AliKarar, M. A smart home energy management system using IoT and big data analytics approach. *IEEE Trans. Consum. Electron.* **2017**, *63*, 426–434. [CrossRef]
65. Martinez, M.T.V.; Comech, M.P.; Hurtado, A.A.P.; Olivan, M.A.; Corton, D.L.; Del Castillo, C.R. Software-Defined Analog Processing Based on IEC 61850 implemented in an Edge Hardware Platform to be used in Digital Substations. *IEEE Access* **2024**, *12*, 11549–11560. [CrossRef]
66. Ahmed, M.A.; Kim, Y.C. Hierarchical communication network architectures for offshore wind power farms. *Energies* **2014**, *7*, 3420–3437. [CrossRef]
67. Liu, F.; Kibalya, G.; Santhosh Kumar, S.; Zhang, P. Challenges of traditional networks and development of programmable networks. In *Software Defined Internet of Everything*; Springer: Cham, Switzerland, 2021; pp. 37–61.
68. Jammal, M.; Singh, T.; Shami, A.; Asal, R.; Li, Y. Software defined networking: State of the art and research challenges. *Comput. Netw.* **2014**, *72*, 74–98. [CrossRef]
69. Baktir, A.C.; Ozgovde, A.; Ersoy, C. How can edge computing benefit from software-defined networking: A survey, use cases, and future directions. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 2359–2391. [CrossRef]
70. Ali, J.; Lee, G.M.; Roh, B.H.; Ryu, D.K.; Park, G. Software-defined networking approaches for link failure recovery: A survey. *Sustainability* **2020**, *12*, 4255. [CrossRef]
71. Bonfim, M.S.; Dias, K.L.; Fernandes, S.F. Integrated NFV/SDN architectures: A systematic literature review. *ACM Computi. Surv. (CSUR)* **2019**, *51*, 1–39. [CrossRef]
72. Greene, K. TR10: Software-Defined Networking. 2009. Available online: https://www.technologyreview.com/2009/02/24/95209/tr10-software-defined-networking/ (accessed on 28 April 2024).
73. Kreutz, D.; Ramos, F.M.; Verissimo, P.E.; Rothenberg, C.E.; Azodolmolky, S.; Uhlig, S. Software-defined networking: A comprehensive survey. *Proc. IEEE* **2014**, *103*, 14–76. [CrossRef]

74. Anerousis, N.; Chemouil, P.; Lazar, A.A.; Mihai, N.; Weinstein, S.B. The origin and evolution of open programmable networks and SDN. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 1956–1971. [CrossRef]
75. Casado, M.; McKeown, N.; Shenker, S. From ethane to SDN and beyond. *ACM SIGCOMM Comput. Commun. Rev.* **2019**, *49*, 92–95. [CrossRef]
76. Siddiqui, S.; Hameed, S.; Shah, S.A.; Ahmad, I.; Aneiba, A.; Draheim, D.; Dustdar, S. Towards Software-Defined Networking-Based IoT Frameworks: A Systematic Literature Review, Taxonomy, Open Challenges and Prospects. *IEEE Access* **2022**, *10*, 70850–70901. [CrossRef]
77. Marschke, D.; Doyle, J.; Moyer, P. *Software Defined Networking (SDN): Anatomy of OpenFlow Volume I*; Lulu Press: Raleigh, NC, USA, 2015; Volume 1.
78. Sur, D.; Pfaff, B.; Ryzhyk, L.; Budiu, M. Full-stack SDN. In Proceedings of the 21st ACM Workshop on Hot Topics in Networks, Austin, TX, USA, 14–15 November 2022; pp. 130–137.
79. Hu, D.; Li, S.; Xue, N.; Chen, C.; Ma, S.; Fang, W.; Zhu, Z. Design and demonstration of SDN-based flexible flow converging with protocol-oblivious forwarding (POF). In Proceedings of the 2015 IEEE Global Communications Conference (GLOBECOM), San Diego, CA, USA, 6–10 December 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 1–6.
80. Ahmad, S.; Mir, A.H. SDN Interfaces: Protocols, taxonomy and challenges. *Int. J. Wirel. Microwave Technol.* **2022**, *12*, 11–32. [CrossRef]
81. Bianchi, G.; Bonola, M.; Capone, A.; Cascone, C. Openstate: Programming platform-independent stateful openflow applications inside the switch. *ACM SIGCOMM Comput. Commun. Rev.* **2014**, *44*, 44–51. [CrossRef]
82. Belter, B.; Binczewski, A.; Dombek, K.; Juszczyk, A.; Ogrodowczyk, L.; Parniewicz, D.; Stroiński, M.; Olszewski, I. Programmable abstraction of datapath. In Proceedings of the 2014 Third European Workshop on Software Defined Networks, Budapest, Hungary, 1–3 September 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 7–12.
83. Haleplidis, E.; Hadi Salim, J.; Denazis, S.; Koufopavlou, O. Towards a network abstraction model for SDN. *J. Netw. Syst. Manag.* **2015**, *23*, 309–327. [CrossRef]
84. Salman, O.; Elhajj, I.H.; Kayssi, A.; Chehab, A. SDN controllers: A comparative study. In Proceedings of the 2016 18th Mediterranean Electrotechnical Conference (MELECON), Lemesos, Cyprus , 18–20 April 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 1–6.
85. Zhu, L.; Karim, M.M.; Sharif, K.; Xu, C.; Li, F.; Du, X.; Guizani, M. SDN controllers: A comprehensive analysis and performance evaluation study. *ACM Comput. Surv. (CSUR)* **2020**, *53*, 1–40. [CrossRef]
86. Khattak, Z.K.; Awais, M.; Iqbal, A. Performance evaluation of OpenDaylight SDN controller. In Proceedings of the 2014 20th IEEE international Conference on Parallel and Distributed Systems (ICPADS), Hsinchu, Taiwan , 16–19 December 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 671–676.
87. Berde, P.; Gerola, M.; Hart, J.; Higuchi, Y.; Kobayashi, M.; Koide, T.; Lantz, B.; O'Connor, B.; Radoslavov, P.; Snow, W.; et al. ONOS: Towards an open, distributed SDN OS. In Proceedings of the 3rd Workshop on Hot Topics in Software Defined Networking, Chicago, IL, USA, 22 August 2014; pp. 1–6.
88. Prete, L.R.; Shinoda, A.A.; Schweitzer, C.M.; De Oliveira, R.L.S. Simulation in an SDN network scenario using the POX Controller. In Proceedings of the 2014 IEEE Colombian Conference on Communications and Computing (COLCOM), Bogota, Colombia, 4–6 June 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 1–6.
89. Morales, L.V.; Murillo, A.F.; Rueda, S.J. Extending the floodlight controller. In Proceedings of the 2015 IEEE 14th International Symposium on Network Computing and Applications, Cambridge, MA, USA, 28–30 September 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 126–133.
90. Priyadarsini, M.; Bera, P.; Bampal, R. Performance analysis of software defined network controller architecture—A simulation based survey. In Proceedings of the 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, India , 22–24 March 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1929–1935.
91. Sheikh, M.N.A.; Halder, M.; Kabir, S.S.; Miah, M.W.; Khatun, S. SDN-Based approach to evaluate the best controller: Internal controller NOX and external controllers POX, ONOS, RYU. *Glob. J. Comput. Sci. Technol.* **2019**, *19*, 21–32. [CrossRef]
92. Shah, S.A.; Faiz, J.; Farooq, M.; Shafi, A.; Mehdi, S.A. An architectural evaluation of SDN controllers. In Proceedings of the 2013 IEEE International Conference on Communications (ICC), Budapest, Hungary, 9–13 June 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 3504–3508.
93. Lee, B.; Park, S.H.; Shin, J.; Yang, S. IRIS: The Openflow-based recursive SDN controller. In Proceedings of the 16th International Conference on Advanced Communication Technology, Pyeongchang, Republic of Korea, 16–19 February 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 1227–1231.
94. Quincozes, S.E.; Soares, A.A.Z.; Oliveira, W.; Cordeiro, E.B.; Lima, R.A.; Muchaluat-Saade, D.C.; Ferreira, V.C.; Lopes, Y.; Vieira, J.L.; Uchôa, L.M.; et al. Survey and Comparison of SDN Controllers for Teleprotection and Control Power Systems. In Proceedings of the 9th Latin American Network Operations and Management Symposium (LANOMS), Niteroi, Brazil, 25–27 September 2019.
95. Isong, B.; Molose, R.R.S.; Abu-Mahfouz, A.M.; Dladlu, N. Comprehensive review of SDN controller placement strategies. *IEEE Access* **2020**, *8*, 170070–170092. [CrossRef]
96. Benzekki, K.; El Fergougui, A.; Elbelrhiti Elalaoui, A. Software-defined networking (SDN): A survey. *Secur. Commun. Netw.* **2016**, *9*, 5803–5833. [CrossRef]

97. Rowshanrad, S.; Abdi, V.; Keshtgari, M. Performance evaluation of SDN controllers: Floodlight and OpenDaylight. *IIUM Eng. J.* **2016**, *17*, 47–57. [CrossRef]
98. Feamster, N.; Rexford, J.; Zegura, E. The road to SDN: An intellectual history of programmable networks. *ACM SIGCOMM Comput. Commun. Rev.* **2014**, *44*, 87–98. [CrossRef]
99. Li, G.; Wang, X.; Zhang, Z. SDN-based load balancing scheme for multi-controller deployment. *IEEE Access* **2019**, *7*, 39612–39622. [CrossRef]
100. Stallings, W. *Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud*; Addison-Wesley Professional: Reading, MA, USA, 2015.
101. Zhu, L.; Karim, M.M.; Sharif, K.; Li, F.; Du, X.; Guizani, M. SDN controllers: Benchmarking & performance evaluation. *arXiv* **2019**, arXiv:1902.04491.
102. Mamushiane, L.; Lysko, A.; Dlamini, S. A comparative evaluation of the performance of popular SDN controllers. In Proceedings of the 2018 Wireless Days (WD), Dubai, United Arab Emirates, 3–5 April 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 54–59.
103. Islam, S.; Khan, M.A.I.; Shorno, S.T.; Sarker, S.; Siddik, M.A. Performance evaluation of SDN controllers in wireless network. In Proceedings of the 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), Dhaka, Bangladesh , 3–5 May 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–5.
104. Singh, A.; Kaur, N.; Kaur, H. Extensive performance analysis of OpenDayLight (ODL) and Open Network Operating System (ONOS) SDN controllers. *Microprocess. Microsyst.* **2022**, *95*, 104715. [CrossRef]
105. Ersue, M. ETSI NFV management and orchestration-An overview. In Proceedings of the 88th IETF, Vancouver, BC, Canada, 13 November 2013.
106. Huang, Y.X.; Chou, J. A survey of NFV network acceleration from ETSI perspective. *Electronics* **2022**, *11*, 1457. [CrossRef]
107. Alam, I.; Sharif, K.; Li, F.; Latif, Z.; Karim, M.M.; Biswas, S.; Nour, B.; Wang, Y. A survey of network virtualization techniques for Internet of Things using SDN and NFV. *ACM Comput. Surv.* **2020**, *53*, 1–40. [CrossRef]
108. Sun, G.; Xu, Z.; Yu, H.; Chen, X.; Chang, V.; Vasilakos, A.V. Low-latency and resource-efficient service function chaining orchestration in network function virtualization. *IEEE Internet Things J.* **2019**, *7*, 5760–5772. [CrossRef]
109. Blenk, A.; Basta, A.; Reisslein, M.; Kellerer, W. Survey on network virtualization hypervisors for software defined networking. *IEEE Commun. Surv. Tutor.* **2015**, *18*, 655–685. [CrossRef]
110. Gjermundrod, H.; Bakken, D.E.; Hauser, C.H.; Bose, A. GridStat: A flexible QoS-managed data dissemination framework for the power grid. *IEEE Trans. Power Deliv.* **2008**, *24*, 136–143. [CrossRef]
111. Zopellaro Soares, A.A.; Lucas Vieira, J.; Quincozes, S.E.; Ferreira, V.C.; Uchôa, L.M.; Lopes, Y.; Passos, D.; Fernandes, N.C.; Monteiro Moraes, I.; Muchaluat-Saade, D.; et al. SDN-based teleprotection and control power systems: A study of available controllers and their suitability. *Int. J. Netw. Manag.* **2021**, *31*, e2112. [CrossRef]
112. Mo, J. *Performance Modeling of Communication Networks with Markov Chains*; Springer Nature: Cham, Swizerland, 2022.
113. Quy, V.K.; Nam, V.H.; Linh, D.M.; Ban, N.T.; Han, N.D. A survey of QoS-aware routing protocols for the MANET-WSN convergence scenarios in IoT networks. *Wirel. Pers. Commun.* **2021**, *120*, 49–62. [CrossRef]
114. Curado, M.; Monteiro, E. A survey of QoS routing algorithms. In Proceedings of the International Conference on Information Technology (ICIT 2004), Istanbul, Turkey, 17–19 December 2004.
115. Paul, P.; Raghavan, S. Survey of QoS routing. In Proceedings of the International Conference on Computer Communication, New York, NY, USA, 23–27 June 2002; Volume 15, p. 50.
116. Wang, W.; Xu, Y.; Khanna, M. A survey on the communication architectures in smart grid. *Comput. Netw.* **2011**, *55*, 3604–3629. [CrossRef]
117. Ren, L.; Qin, Y.; Wang, B.; Zhang, P.; Luh, P.B.; Jin, R. Enabling resilient microgrid through programmable network. *IEEE Trans. Smart Grid* **2016**, *8*, 2826–2836. [CrossRef]
118. Zhang, D.; Li, C.; Goh, H.H.; Ahmad, T.; Zhu, H.; Liu, H.; Wu, T. A comprehensive overview of modeling approaches and optimal control strategies for cyber-physical resilience in power systems. *Renew. Energy* **2022**, *189*, 1383–1406. [CrossRef]
119. Sahay, R.; Blanc, G.; Zhang, Z.; Debar, H. Towards autonomic DDoS mitigation using software defined networking. In Proceedings of the SENT 2015: NDSS Workshop on Security of Emerging Networking Technologies, San Diego, CA, USA, 8–11 February 2015.
120. Sahay, R.; Meng, W.; Jensen, C.D. The application of Software Defined Networking on securing computer networks: A survey. *J. Netw. Comput. Appl.* **2019**, *131*, 89–108. [CrossRef]
121. Zhu, R.; Liu, C.C.; Hong, J.; Wang, J. Intrusion detection against MMS-based measurement attacks at digital substations. *IEEE Access* **2020**, *9*, 1240–1249. [CrossRef]
122. Presekal, A.; Ştefanov, A.; Rajkumar, V.S.; Palensky, P. Attack Graph Model for Cyber-Physical Power Systems using Hybrid Deep Learning. *IEEE Trans. Smart Grid* **2023**, *14*, 4007–4020. [CrossRef]
123. IEA. *Power Systems in Transition—Analysis*; IEA: Paris, France, 2021.
124. Goud, K.S.; Gidituri, S.R. Security challenges and related solutions in software defined networks: A survey. *Int. J. Comput. Netw. Appl.* **2022**, *9*, 22–37. [CrossRef] [PubMed]
125. Onyeji, I.; Bazilian, M.; Bronk, C. Cyber security and critical energy infrastructure. *Electr. J.* **2014**, *27*, 52–60. [CrossRef]
126. Rajkumar, V.S.; Ştefanov, A.; Presekal, A.; Palensky, P.; Torres, J.L.R. Cyber Attacks on Power Grids: Causes and Propagation of Cascading Failures. *IEEE Access* **2023**, *11*, 103154–103176. [CrossRef]
127. Assante, M.J.; Lee, R.M. The industrial control system cyber kill chain. *SANS Inst. InfoSec Read. Room* **2015**, *1*, 2.

128. EnergiCERT. *Cyber Attacks against European Energy & Utility Companies TLP: Clear*; EnergiCERT: Harrogate, UK, 2022.
129. Mohan, A.M.; Meskin, N.; Mehrjerdi, H. A comprehensive review of the cyber-attacks and cyber-security on load frequency control of power systems. *Energies* **2020**, *13*, 3860. [CrossRef]
130. Arghandeh, R.; Von Meier, A.; Mehrmanesh, L.; Mili, L. On the definition of cyber-physical resilience in power systems. *Renew. Sustain. Energy Rev.* **2016**, *58*, 1060–1069. [CrossRef]
131. Poulin, C.; Kane, M.B. Infrastructure resilience curves: Performance measures and summary metrics. *Reliab. Eng. Syst. Saf.* **2021**, *216*, 107926. [CrossRef]
132. Shamugam, V.; Murray, I.; Leong, J.; Sidhu, A.S. Software Defined Networking challenges and future direction: A case study of implementing SDN features on OpenStack private cloud. In *Proceedings of the IOP Conference Series: Materials Science and Engineering*; IOP Publishing: Bristol, UK, 2016; Volume 121, p. 012003.
133. Kabashkin, I. Availability of Services in Wireless Sensor Network with Aerial Base Station Placement. *J. Sens. Actuator Netw.* **2023**, *12*, 39. [CrossRef]
134. Pérez-Rúa, J.A.; Lumbreras, S.; Ramos, A.; Cutululis, N.A. Reliability-based topology optimization for offshore wind farm collection system. *Wind Energy* **2022**, *25*, 52–70. [CrossRef]
135. Abdukhakimov, A.; Bhardwaj, S.; Gashema, G.; Kim, D.S. Reliability analysis in smart grid networks considering distributed energy resources and storage devices. *Int. J. Electr. Electron. Eng. Telecommun.* **2019**, *8*, 233–237. [CrossRef]
136. Cisco, U. *Cisco Annual Internet Report (2018–2023)*; White Paper; Cisco: San Jose, CA, USA, 2020.
137. Chu, C.Y.; Xi, K.; Luo, M.; Chao, H.J. Congestion-aware single link failure recovery in hybrid SDN networks. In Proceedings of the 2015 IEEE Conference on Computer Communications (INFOCOM), Hong Kong, China, 26 April–1 May 2015; IEEE 2015; pp. 1086–1094.
138. Saleh, Z.Z.; Qadir, Q.M. The Downside of Software-Defined Networking in Wireless Network. *UKH J. Sci. Eng.* **2020**, *4*, 147–156. [CrossRef]
139. Lu, Z.; Sun, C.; Cheng, J.; Li, Y.; Li, Y.; Wen, X. SDN-enabled communication network framework for energy Internet. *J. Comput. Netw. Commun.* **2017**, *2017*, 8213854. [CrossRef]
140. Aly, W.H.F.; Kanj, H.; Alabed, S.; Mostafa, N.; Safi, K. Dynamic Feedback versus Varna-Based Techniques for SDN Controller Placement Problems. *Electronics* **2022**, *11*, 2273. [CrossRef]
141. Singh, A.K.; Maurya, S.; Kumar, N.; Srivastava, S. Heuristic approaches for the reliable SDN controller placement problem. *Trans. Emerg. Telecommun. Technol.* **2020**, *31*, e3761. [CrossRef]
142. Hohlfeld, O.; Kempf, J.; Reisslein, M.; Schmid, S.; Shah, N. Guest editorial scalability issues and solutions for software defined networks. *IEEE J. Sel. Areas Commun.* **2018**, *36*, 2595–2602. [CrossRef]
143. Ahmad, S.; Mir, A.H. Scalability, consistency, reliability and security in SDN controllers: A survey of diverse SDN controllers. *J. Netw. Syst. Manag.* **2021**, *29*, 1–59. [CrossRef]
144. Madni, A.M.; Erwin, D.; Sievers, M. Constructing models for systems resilience: Challenges, concepts, and formal methods. *Systems* **2020**, *8*, 3. [CrossRef]
145. Goswami, B.; Hu, S.; Feng, Y. Software-defined networking for real-time network systems. In *Handbook of Real-Time Computing*; Springer: Singapore, 2022; pp. 935–959.
146. Uddin, M.; Mukherjee, S.; Chang, H.; Lakshman, T. SDN-based service automation for IoT. In Proceedings of the 2017 IEEE 25th International Conference on Network Protocols (ICNP), Toronto, ON, Canada, 10–13 October 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–10.
147. Gyllstrom, D.; Braga, N.; Kurose, J. Recovery from link failures in a smart grid communication network using openflow. In Proceedings of the 2014 IEEE International Conference on Smart Grid Communications (SmartGridComm), Venice, Italy, 3–6 November 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 254–259.
148. Petale, S.; Thangaraj, J. Link failure recovery mechanism in software defined networks. *IEEE J. Sel. Areas Commun.* **2020**, *38*, 1285–1292. [CrossRef]
149. Miura, H.; Hirata, K.; Tachibana, T. P4-based design of fast failure recovery for software-defined networks. *Comput. Netw.* **2022**, *216*, 109274. [CrossRef]
150. Bardsiri, A.K.; Hashemi, S.M. Qos metrics for cloud computing services evaluation. *Int. J. Intell. Syst. Appl.* **2014**, *6*, 27. [CrossRef]
151. Guo, Y.; Wang, W.; Zhang, H.; Guo, W.; Wang, Z.; Tian, Y.; Yin, X.; Wu, J. Traffic engineering in hybrid software defined network via reinforcement learning. *J. Netw. Comput. Appl.* **2021**, *189*, 103116. [CrossRef]
152. Keshari, S.K.; Kansal, V.; Kumar, S. A systematic review of quality of services (QoS) in software defined networking (SDN). *Wirel. Pers. Commun.* **2021**, *116*, 2593–2614. [CrossRef]
153. Bakshi, K. Considerations for software defined networking (SDN): Approaches and use cases. In Proceedings of the 2013 IEEE Aerospace Conference, Big Sky, MT, USA, 2–9 March 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 1–9.
154. Muhammad, T. Overlay Network Technologies in SDN: Evaluating Performance and Scalability of VXLAN and GENEVE. *Int. J. Comput. Sci. Technol.* **2021**, *5*, 39–75.
155. de Sousa, N.F.S.; Rothenberg, C.E. CLARA: Closed loop-based zero-touch network management framework. In Proceedings of the 2021 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), Heraklion, Greece, 9–11 November 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 110–115.

156. ETSI GS ZSM 002. Zero-Touch Network and Service Management (ZSM); Reference Architecture. 2019. Available online: https://www.etsi.org/deliver/etsi_gs/ZSM/001_099/002/01.01.01_60/gs_ZSM002v010101p.pdf (accessed on 28 April 2024).
157. Khan, T.A.; Muhammad, A.; Abbas, K.; Song, W.C. Intent-based networking platform: An automated approach for policy and configuration of next-generation networks. In Proceedings of the 36th Annual ACM Symposium on Applied Computing, Virtual Event, 22–26 March 2021; pp. 1921–1930.
158. Mehmood, K.; Mendis, H.K.; Kralevska, K.; Heegaard, P.E. Intent-based network management and orchestration for smart distribution grids. In Proceedings of the 2021 28th International Conference on Telecommunications (ICT), London, UK, 1–3 June 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–6.
159. Martini, B.; Gharbaoui, M.; Castoldi, P. Intent-based zero-touch service chaining layer for software-defined edge cloud networks. *Comput. Netw.* **2022**, *212*, 109034. [CrossRef]
160. Velasco, L.; Signorelli, M.; De Dios, O.G.; Papagianni, C.; Bifulco, R.; Olmos, J.J.V.; Pryor, S.; Carrozzo, G.; Schulz-Zander, J.; Bennis, M.; et al. End-to-end intent-based networking. *IEEE Commun. Mag.* **2021**, *59*, 106–112. [CrossRef]
161. Chollon, G.; Ayed, D.; Garriga, R.A.; Zarca, A.M.; Skarmeta, A.; Christopoulou, M.; Soussi, W.; Gur, G.; Herzog, U. ETSI ZSM Driven Security Management in Future Networks. In Proceedings of the 2022 IEEE Future Networks World Forum (FNWF), Montreal, QC, Canada, 10–14 October 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 334–339.
162. Moubayed, A.; Shami, A.; Al-Dulaimi, A. On end-to-end intelligent automation of 6G networks. *Future Internet* **2022**, *14*, 165. [CrossRef]
163. Mishra, R.; Gijare, V.; Malik, S. Zero touch network: A comprehensive network design approach. *Int. J. Eng. Res. Technol.* **2019**, *8*, 792–794.
164. Hyder, M.F.; Fatima, T. Towards crossfire distributed denial of service attack protection using intent-based moving target defense over software-defined networking. *IEEE Access* **2021**, *9*, 112792–112804. [CrossRef]
165. Medvetskyi, M.; Beshley, M.; Klymash, M. A quality of experience management method for intent-based software-defined networks. In Proceedings of the 2021 IEEE 16th International Conference on the Experience of Designing and Application of CAD Systems (CADSM), Lviv, Ukraine, 22–26 February 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 59–62.
166. Coronado, E.; Behravesh, R.; Subramanya, T.; Fernàndez-Fernàndez, A.; Siddiqui, M.S.; Costa-Pérez, X.; Riggio, R. Zero touch management: A survey of network automation solutions for 5G and 6G networks. *IEEE Commun. Surv. Tutor.* **2022**, *24*, 2535–2578. [CrossRef]
167. Rizwan, A.; Jaber, M.; Filali, F.; Imran, A.; Abu-Dayya, A. A zero-touch network service management approach using AI-enabled CDR analysis. *IEEE Access* **2021**, *9*, 157699–157714. [CrossRef]
168. Gallego-Madrid, J.; Sanchez-Iborra, R.; Ruiz, P.M.; Skarmeta, A.F. Machine learning-based zero-touch network and service management: A survey. *Digital Commun. Netw.* **2022**, *8*, 105–123. [CrossRef]
169. Boškov, I.; Yetgin, H.; Vučnik, M.; Fortuna, C.; Mohorčič, M. Time-to-provision evaluation of IoT devices using automated zero-touch provisioning. In Proceedings of the GLOBECOM 2020-2020 IEEE Global Communications Conference, Taipei, Taiwan, 7–11 December 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–7.
170. Dake, D.K.; Gadze, J.D.; Klogo, G.S.; Nunoo-Mensah, H. Multi-agent reinforcement learning framework in sdn-iot for transient load detection and prevention. *Technologies* **2021**, *9*, 44. [CrossRef]
171. Passito, A.; Mota, E.; Bennesby, R.; Fonseca, P. AgNOS: A framework for autonomous control of software-defined networks. In Proceedings of the 2014 IEEE 28th International Conference on Advanced Information Networking and Applications, Victoria, BC, Canada, 13–16 May 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 405–412.
172. Yao, H.; Mai, T.; Xu, X.; Zhang, P.; Li, M.; Liu, Y. NetworkAI: An intelligent network architecture for self-learning control strategies in software defined networks. *IEEE Internet Things J.* **2018**, *5*, 4319–4327. [CrossRef]
173. Hu, Y.; Li, Z.; Lan, J.; Wu, J.; Yao, L. EARS: Intelligence-driven experiential network architecture for automatic routing in software-defined networking. *China Commun.* **2020**, *17*, 149–162. [CrossRef]
174. Casas-Velasco, D.M.; Rendon, O.M.C.; da Fonseca, N.L. Intelligent routing based on reinforcement learning for software-defined networking. *IEEE Trans. Netw. Serv. Manag.* **2020**, *18*, 870–881. [CrossRef]